



universidad
de león



Escuela de Ingenierías Industrial, Informática y Aeroespacial

GRADO EN INGENIERÍA EN ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

Trabajo de Fin de Grado

CONTROL REMOTO DE EQUIPOS DEL LABORATORIO
REMOTE CONTROL OF LABORATORY EQUIPMENT

Autor: Laura González Llamas
Tutor: Natalia Prieto Fernández

(Septiembre, 2022)

UNIVERSIDAD DE LEÓN
Escuela de Ingenierías Industrial, Informática y Aeroespacial
GRADO EN INGENIERÍA EN ELECTRÓNICA INDUSTRIAL Y
AUTOMÁTICA
Trabajo de Fin de Grado

ALUMNO: Laura González Llamas

TUTOR: Natalia Prieto Fernández

TÍTULO: Control remoto de equipos del laboratorio

TITLE: Remote control of laboratory equipment

CONVOCATORIA: Septiembre, 2022

RESUMEN: Este proyecto consiste en la realización de dos interfaces gráficas para controlar equipos de laboratorio de forma remota desde un ordenador. Para ello se han desarrollado dos programas usando el lenguaje de programación Python.

Mediante estas interfaces se podrán controlar tres equipos electrónicos digitales, un generador de señales, una fuente de alimentación y un osciloscopio. Con estas se podrá mandar datos a los instrumentos, así como leerlos a través de un ordenador. En este caso, estos aparatos para los cuales se ha desarrollado el programa, son del fabricante Rigol.

Hoy en día hay un cambio de tendencia en los equipos de laboratorio utilizados para mediciones, ya que se está potenciando el uso de aparatos electrónicos digitales, debido a que son más precisos, tienen mayor resolución y permiten una lectura mucho más rápida que los equipos analógicos. También es importante destacar que los aparatos electrónicos digitales nos permiten almacenar datos.

Estos aparatos pueden ser automatizados, facilitando su lectura, su almacenaje e incluso la forma en que adquieren los datos.

Para la comprobación de la automatización llevada a cabo, se han realizado dos montajes diferentes que convierten la señal de continua a continua. Para ello se han utilizado reguladores de tensión de conmutación.

ABSTRACT: This project consists of the creation of two graphic interfaces to control laboratory equipment remotely from a computer. To this end, two programs have been developed using the Python programming language.

These interfaces can be used to control three digital electronic devices, a signal generator, a power supply and an oscilloscope. These can be used to send data to the instruments as well as to read them out via a computer. In this case, these devices, for which the programme has been developed, are from the manufacturer Rigol.

Nowadays there is a change of trend in the laboratory equipment used for measurements, as the use of digital electronic devices is being promoted, due to the fact that they are more precise, have a higher resolution and allow a much faster reading than analogue equipment. It is also important to note that digital electronics allow us to store data.

These devices can be automated, facilitating their reading, their storage and even the way in which they acquire data.

In order to test the automation carried out, two different assemblies have been made to convert the signal from DC to DC. Switching voltage regulators were used for this purpose.

Palabras clave: Automatización, Interfaz, Python, Equipos electrónicos y Software

Firma del alumno:

VºBº Tutor/es:

Resumen

Este proyecto consiste en la realización de dos interfaces gráficas para controlar equipos de laboratorio de forma remota desde un ordenador. Para ello se han desarrollado dos programas usando el lenguaje de programación Python, mediante varias de sus librerías, entre las que cabe destacar la librería PyVisa.

Mediante una de las interfaces se envían datos e instrucciones tanto a una fuente de tensión digital de 3 salidas y con una potencia de hasta 195 W, como a un generador de señales digital que consta de 2 canales independientes, permitiendo el control de ambos equipos.

La segunda interfaz, se desarrolla para el control de un osciloscopio digital de 4 canales, desde esta interfaz se pueden enviar datos al equipo, obtener y almacenar información de retorno y visualizar gráficas con los datos obtenidos.

Los equipos para los cuales se han desarrollado ambas interfaces son del fabricante Rigol, que está especializado en la fabricación de equipos electrónicos de medición, de alta precisión. Estos equipos están dotados de un puerto USB Device que nos permite conectar el equipo al ordenador.

Actualmente hay un cambio de tendencia en los equipos de laboratorio utilizados para mediciones, ya que se está potenciando el uso de aparatos electrónicos digitales, debido a que son más precisos, tienen mayor resolución y permiten una lectura mucho más rápida que los equipos analógicos. También es importante destacar que los aparatos electrónicos digitales nos permiten almacenar datos.

A pesar de que su precio suele ser más elevado, optar por equipos digitales es una buena decisión, ya que además de todas las ventajas citadas anteriormente, estos aparatos pueden ser automatizados, lo cual es de gran importancia ya que facilitará su lectura, su almacenaje e incluso la forma en que adquieren los datos.

Para la comprobación de la automatización llevada a cabo, se han realizado dos montajes diferentes que convierten la señal de DC a DC. Para ello se han utilizado reguladores de tensión de conmutación, variando tanto la tensión de entrada que nos proporciona la fuente, como la frecuencia del generador de señales.

El primero de ellos invierte la señal de salida con respecto a la de entrada.

En el segundo montaje se invierte y se duplica la señal de salida con respecto a la de entrada.

Posteriormente, se ha procedido a la valoración e interpretación de los resultados obtenidos.

Abstract

This project consists of the creation of two graphic interfaces to control laboratory equipment remotely from a computer. To do this, two programs have been developed using the Python programming language, using several of its libraries, including the PyVisa library.

One of the interfaces sends data and instructions to a digital voltage source with 3 outputs and a power of up to 195 W, as well as to a digital signal generator with 2 independent channels, allowing the control of both devices.

The second interface is developed for the control of a 4-channel digital oscilloscope. From this interface, data can be sent to the equipment, feedback information can be obtained and stored and graphs can be displayed with the data obtained.

The equipment for which both interfaces have been developed is from the manufacturer Rigol, which specialises in the manufacture of high-precision electronic measuring equipment. This equipment is equipped with a USB Device port that allows us to connect the equipment to the computer.

There is currently a change of trend in the laboratory equipment used for measurements, as the use of digital electronic devices is being promoted, due to the fact that they are more precise, have a higher resolution and allow a much faster reading than analogue equipment. It is also important to note that digital electronics allow us to store data.

Although their price is usually higher, opting for digital equipment is a good decision, since in addition to all the advantages mentioned above, these devices can be automated, which is of great importance as it will facilitate their reading, storage and even the way in which they acquire data.

In order to test the automation carried out, two different assemblies have been made to convert the signal from DC to DC. To do this, switching voltage

regulators have been used, varying both the input voltage provided by the source and the frequency of the signal generator.

The first of these inverts the output signal with respect to the input signal.

In the second assembly, the output signal is inverted and doubled with respect to the input signal.

Subsequently, the results obtained were evaluated and interpreted.

Índice de contenidos

Resumen	3
Abstract	5
1. Introducción	12
2. Objetivos	13
3. Estado del arte	14
4. Desarrollo	23
4.1. ETAPAS DEL TRABAJO	23
4.2. PLANTEAMIENTO DEL PROBLEMA	24
4.3. MODELADO DEL SISTEMA DE SOLUCIÓN	25
4.4. TECNOLOGÍAS EMPLEADAS	26
4.4.1. LENGUAJE DE PROGRAMACIÓN PYTHON	26
4.4.2. INTERFAZ GRÁFICA (GUI)	29
4.4.3. NI VISA	29
4.4.4. PYVISA	30
4.4.5. COMANDOS SCPI	32
4.4.6. QT DESIGNER	33
4.4.7. FUENTE DE ALIMENTACIÓN	35
4.4.8. GENERADOR DE FUNCIONES	36
4.4.9. OSCILOSCOPIO	37
4.4.10. ICL7660	39
4.5. INTERFAZ DEL PROGRAMA	41
4.6. ENSAYOS REALIZADOS	55
5. Resultados obtenidos	62
6. Conclusiones y recomendaciones	70
Referencias	72

Índice de figuras

Figura 1.1. Interfaz Xerox Star 8010	12
Figura 3.1. Componentes electrónicos	14
Figura 3.2. Montaje automatización de equipos	19
Figura 3.3. Lenguaje de programación COBOL	22
Figura 4.1. Logo de Python	26
Figura 4.2. National Instruments	30
Figura 4.3. Conexión del usuario con el osciloscopio	31
Figura 4.4. Ejemplo comando SCPI y write	32
Figura 4.5. Ejemplo comando SCPI y query	32
Figura 4.6. Aplicación Qt Designer	35
Figura 4.7. Fuente de alimentación similar a la utilizada	36
Figura 4.8. Generador de señales utilizado	37
Figura 4.9. Osciloscopio utilizado	39
Figura 4.10. GUI Generador de señales y fuente de tensión	42
Figura 4.11. GUI Osciloscopio	46
Figura 4.12. Zona fija inicial para GUI de Qt	48
Figura 4.13. Ejemplo de enviar datos a un label y de llamar a una función	50
Figura 4.14. Zona fija final para GUI de Qt	52
Figura 4.15. Comandos SCPI utilizados en ambos programas	55
Figura 4.16. Gráfica para alto valor de frecuencia y 12000 puntos	57
Figura 4.17. Problema de muestreo	59

Figura 5.1. Esquema primer circuito	62
Figura 5.2. Montaje primer circuito	63
Figura 5.3. Interfaz fuente y generador del primer circuito	64
Figura 5.4. Interfaz osciloscopio del primer circuito	64
Figura 5.5. Gráfica obtenida en el primer circuito	65
Figura 5.6. Visualización resultados del primer circuito en el osciloscopio	65
Figura 5.7. Esquema segundo circuito	66
Figura 5.8. Montaje segundo circuito	67
Figura 5.9. Interfaz fuente y generador del segundo circuito	67
Figura 5.10. Interfaz osciloscopio del segundo circuito	68
Figura 5.11. Gráfica obtenida en el segundo circuito	68
Figura 5.12. Visualización resultados del segundo circuito en el osciloscopio	69

Glosario

VISA	Virtual Instrument Software Architecture.
PyVisa	Python Virtual Instrument Software Architecture.
DC	Direct Current
PARC	Palo Alto Research Center
PC	Personal Computer
AC	Alternate Current
SCPI	Standard Commands for Programmable Instruments
ID	Identification
IDE	Integrated Development Environment
PyPI	Python Package Index
GUI	Graphical User Interface
NI	National Instruments
NI MAX	NI Measurement and Automation Explorer
ASCII	American Standard Code for Information Interchange
XML	eXtensible Markup Language
Ui	User Interface Format
CE	Comunidad Europea
TUV	Technischer Überwachungs-Verein
IEC	International Electrotechnical Commission
VDE	Visual Development Environment
EC	European Community

USB	Universal Serial Bus
LAN	Local Area Network
IO	Input/Output
AM	Amplitud Modulada
FM	Frecuencia Modulada
PM	Pulse Modulation
ASK	Amplitude Shift Keying
FSK	Frequency Shift Keying
PSK	Pre Shared Key
PWM	Pulse Width Modulation
RAM	Random Access Memory
CSV	Comma Separated Values

1. Introducción

Hoy en día hay aplicaciones informáticas en cualquier sitio, las cuáles se usan continuamente. Tras la gran mayoría de estas aplicaciones, se esconde una interfaz gráfica que permite al usuario utilizar la aplicación de una forma más sencilla y visual. Una interfaz gráfica se podría decir que es el enlace entre la máquina y el usuario, permitiendo que este utilice las aplicaciones sin saber programación.

Habría que ir hasta la década de los 70 donde en el centro de investigación de la multinacional Xerox, desarrolló las primeras investigaciones sobre una interfaz de usuario [1].

La primera interfaz gráfica llamada PARC User Interface (Interfaz de Usuario PARC), se realizó en dicha multinacional. Esta interfaz mostraba elementos como ventanas, casillas, botones, menús, barras de desplazamiento o el concepto de escritorio para hablar de lo que se ve en pantalla, los cuales hoy en día se siguen utilizando [1].

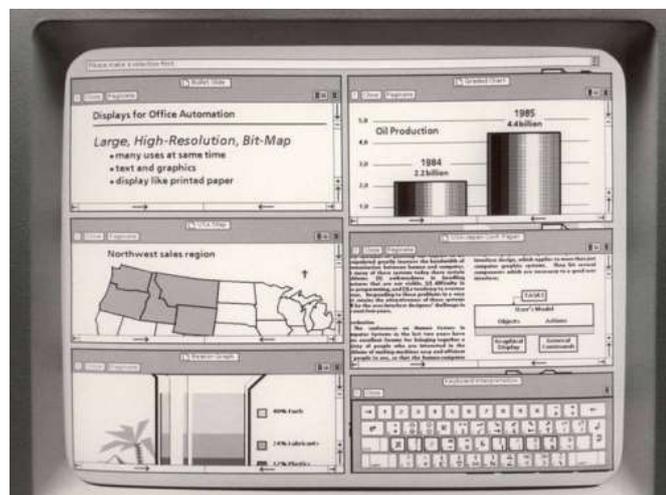


Figura 1.1. Interfaz Xerox Star 8010. (Fuente [1])

Gracias al gran crecimiento e influencia de las interfaces gráficas, los ordenadores han pasado de ser máquinas con una función determinada, para convertirse en máquinas con un montón de funciones, que pueden ser usadas fácilmente por cualquier persona.

2. Objetivos

Este proyecto tiene como objetivo principal la automatización de equipos de laboratorio, esta permitirá una mejor y más precisa lectura de datos, así como un manejo sencillo de los equipos. También proporcionará una rápida adquisición de datos, la cual se podrá consultar en cualquier instante y sin tener que estar el ordenador conectado al equipo.

Los objetivos específicos de este proyecto son:

1. Conectar y gobernar de forma remota la instrumentación al PC. De esta forma se podrá enviar datos al equipo desde el ordenador, a la vez que este recibe y almacena información de los diferentes instrumentos.
2. Realizar interfaces gráficas para el manejo de los equipos. Estas interfaces deben ser sencillas y claras para que todos los usuarios puedan utilizarlas. De esta forma se podrá llegar a un público amplio.
3. Utilizar softwares de programación gratuitos para evitar sobrecostes. Son muchos los lenguajes de programación que a día de hoy se pueden utilizar, pero no todos ellos son gratuitos.
4. Envío de los datos de entrada requeridos por el usuario a los equipos conectados al PC.
5. Captura de los datos en tiempo real y almacenamiento de los mismos, para su posterior estudio.
6. Análisis de los datos, así como la exposición de dichos datos de forma gráfica, permitiendo un rápido análisis del funcionamiento de los circuitos.
7. Comprobación del funcionamiento de la interfaz por medio de un circuito convertidor de DC a DC a través de reguladores de tensión de conmutación.

3. Estado del arte

Un laboratorio electrónico permite probar y construir circuitos y equipos electrónicos, tanto digitales como analógicos, pudiendo medir o adquirir datos de los diferentes montajes que se realicen.

En un laboratorio electrónico se puede encontrar distintos elementos.

En primer lugar, cabe destacar una serie de componentes muy importantes para el diseño de circuitos y los cuales son imprescindibles en cualquier lugar donde se desempeñen labores electrónicas, son los componentes electrónicos, tanto pasivos como activos.

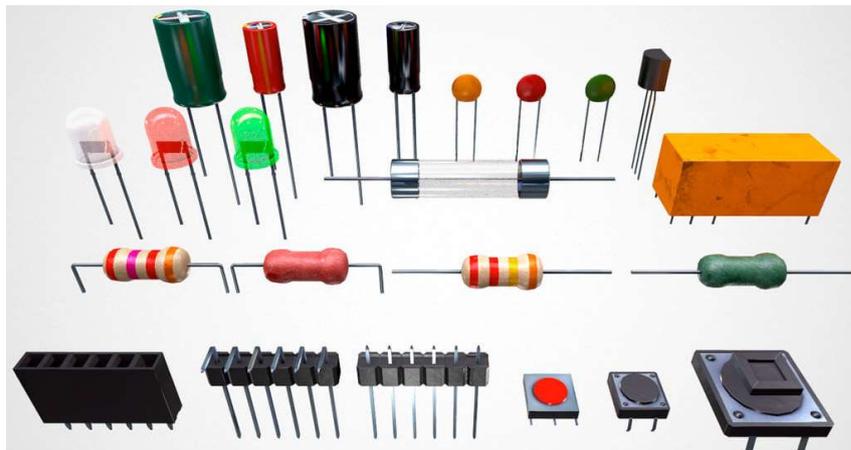


Figura 3.1. Componentes electrónicos. (Fuente: [2])

Dentro de los componentes activos, los más comunes son:

- Diodos: permiten el flujo de corriente en un solo sentido.
- Transistores: se utilizan para entregar una señal de salida en respuesta a una de entrada.
- Circuitos integrados: es lo que popularmente se conoce como chip. Hay diferentes tipos, ya que nos simplifican una función.

Entre los diferentes microchips que existen cabe destacar el regulador de tensión de conmutación. Estos circuitos pueden tomar energía fácilmente de una fuente de tensión de entrada y empujarlos en la salida. Son la mejor opción para un circuito de administración de alimentación. Al recibir una entrada de alto voltaje, antes de disminuirlo, estabiliza la tensión al nivel que se necesite, dependiendo del dispositivo electrónico.

Con estos reguladores se pueden realizar convertidores de corriente continua a corriente continua, los cuales son usados principalmente para almacenar energía en supercondensadores (sirven para estabilizar cargas variantes y proporcionar una carga rápida, o como búfer de energía para suavizar los cambios de tensión) o en baterías (muy importantes porque aparte de almacenar la corriente eléctrica cuando el sistema está apagado, alimentará de corriente eléctrica al sistema, para que este pueda realizar otras funciones).

Por otro lado, los principales componentes electrónicos pasivos son:

-Resistencias: modifican el paso de la corriente.

-Condensadores: almacenan energía en forma de campo eléctrico.

-Bobinas: almacenan energía en forma de campo magnético.

Los componentes que se han visto anteriormente, tanto pasivos como activos, no suelen tener un coste elevado, por lo que su adquisición se puede realizar fácilmente y sin tener mucho presupuesto.

Para colocar todos estos componentes y diseñar así un circuito, se necesitará tanto una placa protoboard como cables para unir los puntos de los diferentes elementos.

Una vez que se tiene el circuito montado, nos interesará o bien proporcionarle una señal de entrada o visualizar las formas de onda que este nos está emitiendo, o simplemente conocer los valores que este posee, e incluso variarlos. Para todo esto se necesitan equipos electrónicos, los más comunes serían:

-Polímetro/multímetro: este aparato permite realizar mediciones de diversas magnitudes eléctricas, esto es posible debido a que están dotados de diferentes

circuitos eléctricos que se selecciona dependiendo de la magnitud eléctrica que se quiera medir y el orden de magnitud de su valor. Las principales mediciones que se pueden hacer con este son: continuidad, resistencia, intensidad y voltaje entre otras [3].

A parte de realizar medidas, este aparato nos permite el ajuste y reparación de otros equipos electrónicos [3].

Los polímetros pueden ser tanto analógicos como digitales, siendo estos últimos los más utilizados debido a su fácil manejo y visualización.

-Fuente de tensión: es dispositivo electrónico que permite suministrar diferentes voltajes para distintas aplicaciones, es decir; con ella se controlan las variables de corriente y voltaje. Hay diferentes tipos según la señal de salida (fijas o regulables), según si son digitales o analógicas, y dependiendo del número de canales de salida que disponga [3].

-Generador de señales: es un equipo electrónico que produce ondas senoidales, cuadradas, triangulares... Además de crear señales TTL. Estas señales nos sirven para ver la reacción de un circuito ante una señal determinada, o bien para saber el rango de frecuencias que es capaz de soportar un circuito, de manera estable [3].

-Osciloscopio: es un equipo electrónico que permite la visualización gráfica de señales eléctricas que varían con el tiempo. Estos dispositivos no sólo se utilizan para la visualización de gráficas, si no que con ellos también se pueden realizar mediciones o localizar averías de un circuito, determinar que parte de la señal es DC y cual AC o incluso saber qué parte de la señal es ruido [3].

Los osciloscopios también los se pueden encontrar analógicos o digitales, siendo los primeros recomendados cuando lo principal es visualizar variaciones rápidas de una señal a tiempo real, mientras que los osciloscopios digitales se utilizan cuando se necesita estudiar y visualizar picos de tensión aleatorios (no repetitivos) [3].

Todos estos son algunos de los aparatos electrónicos disponibles en un laboratorio electrónico. Para conectar estos a nuestro circuito, se necesitarán

sondas BCN y terminales (rojos y negros, para diferenciar el polo positivo y el negativo).

En definitiva, hay infinidad de elementos indispensables en un laboratorio electrónico. Es importante que estos sean de buena calidad ya que, de no ser así, podrían deteriorar piezas o dañarse, ocasionando así complicaciones. También es muy importante que el lugar de trabajo esté completamente limpio, ordenado y organizado, para permitir un buen acceso a los equipos y materiales que se van a utilizar para desarrollar los procesos o investigaciones necesarias. Otro factor a tener en cuenta es el número de tomas de red que se va a poseer, el cual no debería ser menor de 10 ya que prácticamente todos los equipos electrónicos que posee un laboratorio, tienen que estar conectados a la red eléctrica [4].

Gracias al uso de los aparatos electrónicos se puede usar y transmitir de forma sencilla la información de un dispositivo de entrada y mostrarla en un dispositivo de salida [4].

Los equipos electrónicos se dividen en dos tipos, analógicos y digitales. Los primeros trabajan con variables continuas mientras que los segundos lo hacen con variables discretas [3].

La electrónica analógica dominaba durante el siglo XX, esta trabaja con variables continuas, por lo que si se modifica alguna variable se podría llegar a producir un cambio importante en el comportamiento del circuito [5].

Las señales que se producen con la electrónica analógica son señales continuas, ya que representan medidas físicas. Por ello es muy común que los datos se distorsionen, debido al ruido y a las interferencias en la transmisión de estos [6].

Por otro lado, la electrónica digital domina el mercado de computadoras y sistemas inteligentes en el siglo XXI. Esta es la parte de la electrónica que trabaja con las variables discretas, lo que puede implicar una modificación en alguna de las variables del circuito, pero sin cambiar su valor, por lo que no producirá ningún cambio apreciable en el comportamiento del circuito [5].

Las señales que se producen con la electrónica digital son señales discretas de tiempo. Estas, a diferencia con las anteriores, pueden no tener ruido ni interferencias, por lo que, durante la transmisión, los datos recogidos no se deterioran [6].

La diferencia principal entre la electrónica analógica y digital es que en la analógica la información se transmite por medio de pulsos eléctricos y en la digital la transmisión de información es en un formato binario de ceros y unos, donde cada bit es representativo de diferentes amplitudes [6].

Hoy en día la electrónica digital y la analógica coexisten, una muestra de esto es que en un circuito se pueden contener ambas, dependiendo de lo que se quiera analizar o para que se vaya a utilizar este posteriormente.

El sistema analógico ha ido quedando de lado aún a la vista de sus claras ventajas para los profesionales. La industria ha ido empujando la evolución tecnológica hacia la adopción de un sistema digital cada vez más extendido [6].

Es por esto, que con el paso del tiempo la electrónica analógica va a quedar obsoleta, sobre todo, para los aparatos electrónicos, ya que en los últimos años ha habido un importante cambio del formato analógico a la digital [6].

Por otro lado, debido al cambio sufrido en la electrónica, también han ido evolucionando los equipos electrónicos.

Los equipos analógicos son aquellos que emplean numeración mecánica y agujas entre otras características, mientras que los dispositivos digitales prescinden de estos y suelen ser principalmente pantallas digitales y botones.

Como se ha comentado anteriormente, las señales digitales y analógicas no son las mismas, y debido a que las señales son distintas, los instrumentos digitales y analógicos tampoco son iguales.

La instrumentación analógica es aquella que emplea la señal tal cual llega, se utiliza para procesar o calcular una variable. Por otro lado, la instrumentación digital convierte una señal analógica en digital, por lo que no procesa ni calcula una variable de la señal física sino de la señal analógica [7].

En el caso de la instrumentación analógica, existe casi siempre una sola escala de medición, esto hace que los instrumentos analógicos sean intuitivos, por lo que se pueden introducir variables no lineales de forma simple, no como en la instrumentación de electrónica digital, ya que, en esta, al estar sometido a la programación interna del dispositivo, es mucho más difícil introducir una variable no lineal, es decir; hacer una medición específica [7].

Otra diferencia es el cómo se conectan, los instrumentos analógicos no están conectados y esto hace que los procesos de medición y control sean imprecisos (por esto era muy común que antiguamente, las mediciones se hicieran a mano). La instrumentación digital sí que nos permite conectar unos dispositivos con otros debido a que al convertir la señal analógica en información binaria puede procesarla cualquier instrumento. Esto es una de las mayores ventajas de la electrónica digital, esta nos permite un almacenamiento de datos más preciso, rápida y asequible, y si a estos instrumentos se les añade un software de control, las ventajas de esta nueva instrumentación son mucho mayores [7].

Si que es verdad que los equipos digitales suelen ser más costosos que los analógicos, debido a su alta eficacia y eficiencia, así como su manejo sencillo, rapidez y alta resolución.

Durante el siglo XXI de forma simultánea a la electrónica digital, se ha ido avanzando en el uso de la tecnología, lo cuál ha sido un gran apoyo para mejorar estos equipos electrónicos y aportarles nuevas características.

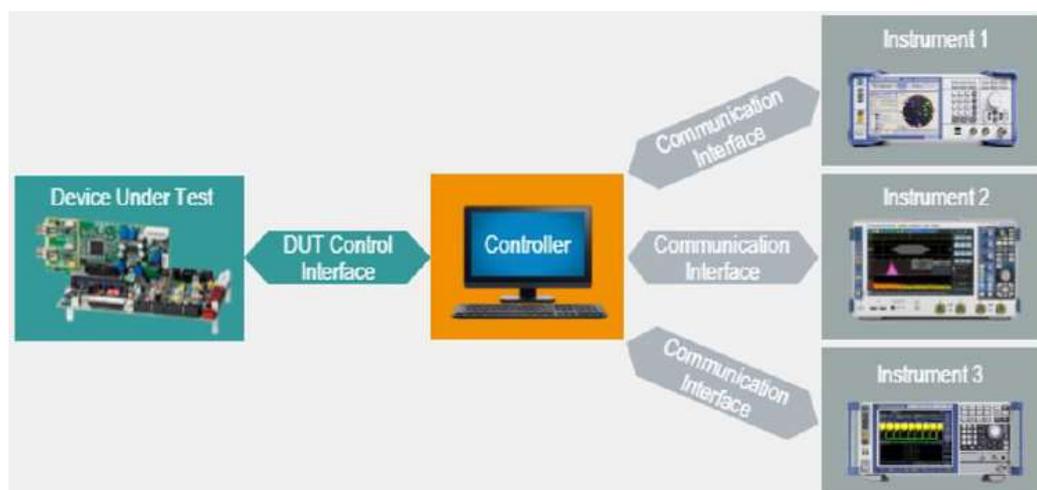


Figura 3.2. Montaje automatización de equipos. (Fuente: [8])

La instrumentación virtual consiste en sustituir equipos físicos por el ordenador, realizando mediciones y manejando el instrumento desde el ordenador. Además, esto nos permite guardar datos en nuestro ordenador y poder consultarlos incluso cuando no se tiene el instrumento electrónico digital.

Esto sólo es posible con aparatos digitales debido a que los analógicos no permiten la creación de un software con el que ser manejados.

Como se puede ver en la Fig.3.2 para realizar la instrumentación virtual se requiere de un controlador (generalmente un ordenador) que contenga un software para el control remoto, mínimo un instrumento electrónico, el cual se quiera controlar y tenga capacidad de control remoto para ser controlado por el ordenador mediante una interfaz. También se puede incorporar un dispositivo bajo prueba, que se comunicará con el ordenador mediante otra interfaz distinta a la anterior, sólo si es necesario.

La instrumentación virtual consiste en reemplazar algunos elementos del hardware de los dispositivos tradicionales por software, mejorando su funcionamiento e incorporando nuevas funcionalidades [9].

Un instrumento virtual tiene una combinación de elementos de software más potente y nuevo hardware, que efectúan las mismas funciones que el instrumento tradicional. Estos elementos aportan al instrumento la capacidad de medición y de control, y añaden las funciones de adquisición, almacenamiento y estudio de los datos si fuera necesario [9].

La instrumentación virtual tiene muchísimas ventajas frente a la instrumentación tradicional, en primer lugar, automatizando los equipos se ahorra tiempo, ya que repetir la misma tarea de medición lleva mucho menos tiempo. Además, se puede operar a distancia, se pueden guardar los datos, y posteriormente realizar cálculos con ellos sin la necesidad de estar conectados al dispositivo a la vez. Los sistemas automatizados son más sencillos de manejar que los aparatos y se pueden expandir a otros equipos con gran facilidad [8].

Además, al tenerlo siempre al alcance, se puede modificar la automatización de los equipos, bastaría con ampliar o modificar el software creado dependiendo de la aplicación que requiera el equipo a utilizar.

La parte más importante de la instrumentación virtual podría considerarse la correcta adquisición de datos. Este es un proceso mediante el que se miden parámetros físicos reales (voltaje, corriente, posición, etc.) y se ingresa en el ordenador para posteriormente ser analizada y procesada. Una vez se almacena, se puede enviar a un sitio remoto o guardarla en el propio ordenador. Para esto se necesita un cable que conecte el dispositivo electrónico al ordenador, como por ejemplo un cable USB device (cable de datos USB) que permite la transmisión de datos de un dispositivo a otro. Con este tipo de cables, aparte de adquirir datos del equipo, también puede recibir datos o señales que le mande el ordenador [10].

La instrumentación virtual ha sufrido una constante evolución desde sus inicios, pero en esta última década, debido a los avances tecnológicos, ha aumentado el estudio e investigación sobre esta nueva instrumentación. Es por ello que en la actualidad se pueden encontrar un montón de aplicaciones de la instrumentación virtual que hace apenas unos años era imposible de imaginar. Poco a poco ira creciendo el número de sistemas que estén basados en esta, así como las áreas de aplicación de estos sistemas [10].

Algo semejante a la instrumentación virtual, y quizás más común, es lo que se conoce por telemando y telemedida.

El telemando consiste en enviar órdenes a distancia a un sistema a través de un medio de transmisión, lo que nos permite controlar sistemas de forma remota, sin que tener por qué estar conectados directamente.

La telemedida consiste en la medición remota de consumos energéticos, sin la necesidad de ir a un contador físico a realizar la lectura de este.

Otro de los grandes avances de esta época ha sido la informática. Dentro de esta, la programación es una pieza clave para cualquier dispositivo a día de hoy, es por eso que actualmente y en los últimos años, la investigación y creación de nuevas técnicas y lenguajes está siendo cada vez mayor.

La programación nos ayuda a establecer una comunicación directa entre los usuarios y el ambiente tecnológico en el que vivimos. Es el medio de comunicación que se tiene con nuestros dispositivos.

4. Desarrollo

4.1. ETAPAS DEL TRABAJO

En primer lugar, se decidió la idea del trabajo en octubre del 2021. Tras barajar diferentes opciones al final se optó por realizar una interfaz para manejar de forma remota el osciloscopio, el generador de funciones y la fuente de tensión.

Después, durante ese mismo mes, se buscó información sobre como programar el software para gobernar estos equipos y se leyó detenidamente la guía de programación que nos facilitaba el fabricante.

En noviembre se procedió a la instalación de Python 3.10 y el paquete NI MAX, ambas son necesarias para programar estos equipos, ya que son compatibles con Python. También se buscó información sobre las librerías que serían necesarias como PyVisa, y posteriormente se instalaron.

Ya en febrero y marzo del 2022 se empezaron a probar los comandos estándar para instrumentos programables, los SCPI. De esta forma se comenzó con la programación. Primero se utilizaron para conocer las ID de los distintos equipos y luego para probar el cómo se comunican los equipos con el ordenador y viceversa.

Durante el mes de abril se hicieron las dos interfaces que son necesarias para el proyecto. Para ello se utilizó el programa QT Designer, ya que facilita el diseño de las de interfaces gráficas.

En mayo se empezó con la programación del software y se continuó con esta en junio. Para programarlo se utilizó Pycharm que es el IDE de Python.

También en junio se empezó a documentar el trabajo con los apartados de introducción y estado del arte.

El mes de julio se empleó para realizar diversas pruebas en el laboratorio y escoger el mejor circuito para verificar el funcionamiento de la automatización y se hicieron fotos y videos del mismo.

Ya en agosto se acabó de documentar el proyecto redactando las partes que faltaban, y dando por concluido el mismo.

4.2. PLANTEAMIENTO DEL PROBLEMA

Los aparatos electrónicos digitales tienen una gran exactitud y una alta precisión. Si es verdad que para mandar datos o capturarlos se tiene que estar pendiente de los equipos y del circuito, y, por ejemplo, no se pueden graficar los datos cuando se quiera si no que tiene que ser a la vez que se están capturando.

Otra ventaja de la automatización es que de esta manera se podría dejar el circuito trabajando y después de horas consultar los datos en el ordenador, ya que este los iría guardando, sin tener que estar todo el rato el usuario al lado del equipo o midiendo los valores del circuito. Es por esto que la automatización de estos equipos nos facilitaría mucho el trabajo.

Si que es verdad que los equipos con los que se ha trabajado, que son del fabricante Rigol, tienen un software propio y que ya está hecho, pero este software tiene inconvenientes y limitaciones. Una de estas limitaciones es que los programas de cada equipo ni están enlazados entre sí ni cumplen con las prestaciones necesarias para el laboratorio de electrónica.

Es por esto que surge la idea de crear un software propio en el que se puedan ejecutar más de un equipo a la vez, y que, aunque se hagan dos programas para controlar los tres equipos, los dos programas se pueden abrir a la vez, es decir; si se ejecuta uno, se puede ejecutar el otro a la vez.

En resumen, la necesidad de llevar a cabo una interfaz gráfica se debe a que resulta difícil y poco intuitivo modificar y adquirir varios datos durante las pruebas que se realicen con los circuitos.

Para realizar automatizar estos equipos primero hay que ver qué lenguajes de programación son compatibles con estos, y si se mira la guía de programación se ve que se puede programar tanto con Python como con Matlab como con C (entre otros).

De todos estos se escoge Python, ya que es un lenguaje de programación gratuito y código abierto con una sintaxis sencilla y multitud de librerías para poder programar fácilmente el software.

4.3. MODELADO DEL SISTEMA DE SOLUCIÓN

Durante este proyecto se han desarrollado dos interfaces gráficas mediante el lenguaje de programación Python. Ambas interfaces nos permiten controlar de forma remota desde un ordenador, equipos electrónicos, haciendo así un uso más sencillo e interactivo. Además, con ellas se podrán configurar y recopilar datos de forma más fácil, así como guardar datos de formas de onda en nuestro equipo para un posible estudio posterior de los mismos.

Una de las interfaces nos va a permitir el control simultáneo de una fuente de tensión digital, modelo DP800 Rigol y un generador de funciones digital, modelo DG1000Z Rigol. La segunda interfaz controla un osciloscopio digital, modelo MSO1000Z Rigol.

Para el manejo de estas, se necesitará un ordenador, así como tres cables USB Device, para conectar cada equipo a nuestro PC,

Para poder realizar este programa, se tendrá que estudiar cada comando que relaciona el programa con el equipo, las diferentes funciones que se quiera que realice la interfaz, y los parámetros y resultados que son de interés para el uso específico de cada aparato electrónico

A continuación, para comprobar el correcto funcionamiento de los programas, se realizarán diferentes pruebas para demostrar la validez de todo lo expuesto.

Para ello, se aplicarán conocimientos de Electrónica de Potencia, con el montaje de un circuito convertidor de corriente continua a corriente continua, utilizando reguladores de tensión de conmutación, concretamente, el modelo ICL7660.

4.4. TECNOLOGÍAS EMPLEADAS

4.4.1. LENGUAJE DE PROGRAMACIÓN PYTHON

Python es el tercer lenguaje de programación más usado en el mundo, es uno de los lenguajes de programación más importantes del momento.

Python fue creado por un programador holandés, Guido van Rossum, a finales de los años 80. Trabajaba como programador en un centro de investigación holandés y en 1989 le dio continuidad al lenguaje de programación ABC, el cual se estaba desarrollando en el mismo centro en el que este estaba trabajando. Se trataba de un lenguaje pensado para principiantes gracias a su fácil aprendizaje y uso, con un código legible, aunque complejo [13].



Figura 4.1. Logo de Python. (Fuente: [14])

Dos años más tarde publicó la primera versión pública de Python, la versión 0.9. Años después publicó la versión 1.0 y la 2.0. Esta última versión incluye la generación de lista, una importante característica de este lenguaje. También fue en esta versión en donde se incorporó el poder hacer referencias cíclicas y así poder recolectar basura dentro del código. Esta versión ya fue desarrollada bajo la dirección de Guido [13].

En 2008 salió la gran última actualización de Python, la versión 3.0. Esta versión solucionaba los fallos de diseño que tenían las anteriores versiones y acumulaba nuevas formas de programar un mismo elemento. La última actualización de la versión 3 de Python fue lanzada a finales del año 2021 [13].

Lo que se muestra en la Fig.4.1 es logo del lenguaje de programación Python.

Este permanece intacto desde la fecha de su creación, lo que le hace ser a Python una entidad reconocible al instante. Si que es verdad que el logo se ha ido mejorando, pero ligeramente, sin grandes cambios [14].

Este logo representa una serpiente de dos colores, basado en los antiguos dibujos mayas. Representan una pitón con cola acortada y cabeza grande. Utiliza una paleta de colores brillantes y formas tradicionales [14].

La sintaxis de Python es muy sencilla y similar al del resto de lenguajes de programación. Además, cuenta con una librería estándar con decenas de módulos, estos cubren las necesidades básicas de cualquier programador.

Python cuenta con un intérprete o consola que permite probar ciertas capacidades del lenguaje sin tener que crear un módulo de este. Incluso, en ocasiones se utiliza como una poderosa calculadora, con capacidades muy similares a las que ofrece el lenguaje Matlab para la realización de ciertas tareas [15].

Una de las características más importantes que tiene Python es que el código escrito en C o en C++ se puede reutilizar. Existen funciones ya establecidas y fáciles de manejar, que permiten no tener que volver a realizar un código ya existente en uno de estos dos lenguajes.

Python es un lenguaje de alto nivel (tienen que ser compilados como lenguajes de bajo nivel antes de ser ejecutados), es por esto que se dice que es un lenguaje de programación interpretado.

Hoy en día, este es el lenguaje de referencia en Data Science y Machine Learning por lo que como se ve, es un lenguaje importante que está en pleno auge.

Como ya se ha dicho, Python cuenta con un montón de librerías, pero cabe destacar que dispone del depósito PyPI. Esta es una herramienta que funciona como un depósito de software para el lenguaje de programación. En esta plataforma se encuentran un montón de paquetes y herramientas muy avanzadas. De esta plataforma salen casi todos los paquetes que se utilizan en los programas informáticos. Un ejemplo de ello es el paquete de instalación de pip, el cual se ha utilizado para instalar todas las bibliotecas necesarias [16].

El comando pip mencionado anteriormente sirve para instalar paquetes de Python. Este comando se ocupa de las dependencias de cada paquete.

Es importante saber que Python es un lenguaje orientado a objetos, la principal filosofía de este es representar elementos que simulen objetos, tal y como los se conocen en la vida real. Estos objetos tendrán unas características, y unas funcionalidades determinadas. Entre los beneficios de que esté orientado a objetos se encuentra el que se pueden crear módulos que llevan a cabo tareas simples y al juntarlos, se resolverían problemas complejos. Otro beneficio es el encapsulamiento, es decir; la ocultación de datos, que permite que los atributos de un objeto pueden ocultarse para que no sean accedidos desde fuera de la definición de su clase [17].

Las principales ventajas de usar Python son: es un lenguaje de alto nivel, polivalente y de paradigmas (se puede usar para un montón de propósitos diferentes), tiene un montón de librerías y módulos integrados, es compatible con todos los sistemas operativos y es fácil de entender. Además, es gratis y de licencia bajo código abierto.

Una vez se sabe el funcionamiento del lenguaje de programación elegido, llega el momento de hablar de GUI.

4.4.2. INTERFAZ GRÁFICA (GUI)

Una interfaz gráfica de usuario (GUI) permite a los distintos usuarios saber fácilmente el funcionamiento de los programas.

Los símbolos son fáciles de comprender y el texto que le acompaña nos explica el programa.

Con una GUI la señal del equipo se transmite al ordenador y este la traduce para realizar al instante un movimiento (ya sea a través del ratón o del teclado) a la pantalla.

Una GUI sirve de traductor entre el usuario y el equipo, utilizando una serie de comandos para llamar al programa.

Una GUI combina el diseño visual y las funciones de programación. Ofrece la posibilidad de añadir botones, menú, campos de navegación y de búsqueda, iconos y widgets, entre otros. Cuando se desarrolla una GUI, hay que usar la programación basada en eventos, ya que las acciones del usuario no se pueden predecir. Es por esto que las GUI no se pueden programar linealmente, sino que deben escribirse de tal manera que un comando solo se ejecute cuando el usuario dé la señal para este [18].

Ahora que se sabe qué es una GUI y se ha elegido el lenguaje de programación, una vez instalado este, lo siguiente que se debe hacer es instalar la aplicación NI Visa.

4.4.3. NI VISA

NI Visa es un controlador de instrumentos NI, una ejecución del estándar VISA. Con este se podrán programar sistemas de instrumentación para interfaces. Los instrumentos que puede controlar pueden ser Ethernet/LXI, GPIB, seriales, USB, PXI y VXI.

En este proyecto se ha optado por instalar NI MAX, ya que esta contiene NI VISA, pero también incluye NI DAQmx (utilizado para la adquisición de datos y acondicionamiento de señales) y otros programas semejantes, por lo que es mucho más completa.

NI MAX es una aplicación gratuita que proporciona acceso al hardware de NI y nos instala todos los controladores citados anteriormente.

La citada compañía National Instruments fue la que introdujo el concepto de ingeniería virtual, citado anteriormente, con el objetivo de desarrollar un software para que un ordenador funcionase de elemento de mediciones.

Hoy en día, NI es una empresa multinacional estadounidense productora de equipos de prueba automatizados y software de instrumentación virtual. Las aplicaciones comunes incluyen adquisición de datos, control de instrumentos y visión artificial [19].



Figura 4.2. National Instruments. (Fuente: [19])

4.4.4. PYVISA

El siguiente paso que se ha realizado es la instalación de la librería PyVISA. Sin esta librería no se podrían programar en Python los tres equipos empleados, ya que los comandos SCPI que nos da el fabricante, pertenecen a esta librería.

PyVISA es una librería disponible para el lenguaje de programación Python. Esta nos permite comunicar el ordenador con todo tipo de equipos de medida.

Programar equipos de mediciones sería muy complicado, debido a que existe una gran cantidad de protocolos que envían y utilizan datos, a través de distintas interfaces y diferentes sistemas de bus, por lo que para simplificar esto, surgió en los años 90 el estándar VISA.

Este estándar configura, programa y soluciona los problemas de los sistemas de medida. Hoy en día se puede encontrar en todos los sistemas operativos y con descarga gratuita [20].

Dentro de este, PyVISA sirve de interfaz para comunicar Python con el estándar VISA. Es un envoltorio de Python para bibliotecas compartidas de VISA, pero también puede servir como interfaz para otras implementaciones de VISA [20].

Lo primero que se debe hacer, una vez instalada la librería PyVISA, es importarla.

Posteriormente se procede a instanciar una clase de tipo `ResourceManager()`.

Para saber la ID de los equipos que se tienen conectados al ordenador, habría que consultar la lista de recursos. Una vez que se verifica que esto funciona y devuelve la ID de los equipos, se va asociando uno a uno el recurso de identificación con el nombre que se le quiere dar al equipo.

Esta es la forma de establecer comunicación entre el usuario y los equipos. Para ellos es indispensable utilizar la librería PyVISA, y es el primer paso de programación a realizar.

```
>>> import pyvisa
>>> rm=pyvisa.ResourceManager()
>>> rm.list_resources()
('USB0::0x1AB1::0x04CE::DS1ZD204401159::INSTR',)
>>> osc = rm.open_resource('USB0::0x1AB1::0x04CE::DS1ZD204401159::INSTR')
```

Figura 4.3. Conexión del usuario con el osciloscopio.

A continuación, hay que ejecutar los comandos SCPI.

4.4.5. COMANDOS SCPI

Los comandos estándar para instrumentos programables definen un estándar para la sintaxis y los comandos para usar el control de los equipos de medición.

Con la llegada de los SCPI se introdujeron comandos genéricos para gobernar cualquier instrumento y definir las clases de estos. Aunque todos ellos los se pueden obtener directamente de la librería PyVISA, estos estándares se agrupan en subsistemas [21].

Los comandos SCPI son cadenas textuales ASCII, que se envían al instrumento a través de la capa física. Los comandos son una serie de una o más palabras clave, muchas de las cuales toman parámetros. A la hora de utilizarlos para programar, estas palabras se pueden utilizar o bien enteras o abreviándolas, cogiendo sólo la parte mayúscula de las mismas. En el caso de coger la palabra entera, hay que respetar el uso de mayúsculas y de minúsculas establecido [21].

Los SCPI pueden realizar una operación establecida (mandar datos al equipo que se está programando), o una consulta (leer datos de ese equipo).

Si lo que se quiere realizar es una consulta, se debe saber que es necesario acompañar a la palabra el signo ' ? ' después de esta.

Casi todos los comandos tienen la opción de consultar o de enviar datos, pero hay algunos que sólo permiten una de las dos opciones. Para saber cuáles tienen una o dos opciones, se debe consultar el manual de estos comandos.

También se pueden usar varios comandos a la vez, para conectarlos sólo es necesario poner entre uno y otro el carácter ' ; ' y entre la acción que se quiere hacer (por ejemplo una medida) y el sobre qué se quiere hacer esa acción (por ejemplo, se quiere medir un voltaje), hay que colocar ' : '.

En Python para saber si lo que se quiere hacer es leer o escribir un mensaje, hay que hacer referencia al equipo sobre el que se quiere hacer las instrucciones, después indicar si se quiere leer (*query*) o escribir (*write*) y ya por último y entre paréntesis, hay que poner el comando SCPI.

```
osc.write(':OUTP 1') #Encender CH1
```

Figura 4.4. Ejemplo comando SCPI y write.

```
osc.query(':FREQ? ') #Consultar el valor de frecuencia
```

Figura 4.5. Ejemplo comando SCPI y query.

Una vez que se sabe conectar al usuario con el equipo empleado e intercambiar información con el mismo, antes de empezar a programar, hay que elegir con qué programa se va a diseñar la interfaz.

Se barajó la opción de realizar el diseño de la interfaz con el paquete de Python Tkinter. Este es una capa orientada a objetos muy sencilla de utilizar para la creación de interfaces. Sin embargo, si se usara este paquete, habría que ir programando uno a uno los botones y las posiciones de estos, así como de las pantallas de lectura.

Es por este inconveniente por lo que se elige usar la aplicación QT Designer como herramienta para la creación del diseño de la interfaz del programa.

4.4.6. QT DESIGNER

Qt Designer pertenece a Qt. Qt empezó siendo una biblioteca de clases, pero en la actualidad es un framework, es decir; un marco de trabajo que ofrece una estructura base para elaborar un proyecto.

Qt Designer es una herramienta de Qt que proporciona una interfaz de usuario en la que lo que se ve es lo que se obtiene. Se utiliza para crear GUI. Es tan fácil de manejar que basta con ir arrastrando y soltando objetos al QWidget (fondo). Nos proporciona tanto botones como etiquetas, así como la opción de crear menús y barras de herramientas [22].

Además, nos ofrece la posibilidad de cambiar funciones, colores, tipos de letra y demás características usando la programación, por lo que se podría tener cada objeto de una forma o color diferente.

Qt Designer es independiente de la plataforma y del lenguaje de programación. El archivo que genera no produce ningún código ni lenguaje de programación en particular, si no que nos crea un archivo .ui. Estos archivos son XML con descripciones cómo generar GUI basadas en Qt [22].

Para transformar este tipo de archivo en un archivo que Python pueda abrir y entender, lo que primero que se tiene que hacer es instalar e importar la librería PyQt5. Esta librería consta de una interfaz de línea de comando para el módulo ui [22].

Una vez es instalada e importada esta librería, se procederá a la transformación del archivo ui en un archivo .py.

Para toda la programación del proyecto se ha utilizado el entorno de desarrollo integrado para la programación en Python llamado PyCharm.

En PyCharm se tiene la posibilidad de instalar directamente las librerías, sin tener que usar el comando *pip* y además se opta de una terminal propia en la cual se pueden ejecutar los comandos.

Para transformar el archivo ui en primer lugar hay que tener en la misma carpeta donde se va a hacer programa el archivo que se quiere pasar a formato .py. Una vez que está este en la misma carpeta, desde la misma terminal del PyCharm se podrá la siguiente línea de código: `pyuic5 -x nombre del archivo.ui -o nombre del archivo.py`.

De esta forma ya se tiene la interfaz gráfica convertida a formato Python.

En este archivo no es aconsejable realizar cambios ni programar la funcionalidad de este, si no que se recomienda crear un archivo .pyw en el que programar el código de funcionamiento del programa. Este archivo nuevo que se crea hay que vincularlo al .py, es decir, hay que importar en este el .py para hacerle referencia y que muestre así la interfaz que ha sido diseñada en este.

A la hora de hablar de Qt hay que comentar una serie de ventajas que tiene respecto a otras formas de realizar interfaces gráficas (como por ejemplo Tkinter en Python). Estas ventajas van desde lo rápido y sencillo que es crear GUI desde Qt Designer, hasta que el propio programa deja ver una vista previa en cualquier momento, de cómo está quedando la interfaz. Además, es un programa que es muy intuitivo y que cualquier persona sin muchos conocimientos de programación, podría usar sin ningún problema.

Por el contrario, la mayor desventaja que tiene el crear el diseño de la interfaz a través del Qt Designer en vez de a través de Tkinter, es que cada vez que se modifica algo del diseño en el Qt, hay que ir al Pycharm y volver a convertirlo en un .py sobrescribiendo el que se tenía. Esto nos llevará poco tiempo, y en su conjunto, es más rápido crear la GUI desde Qt que desde Tkinter.

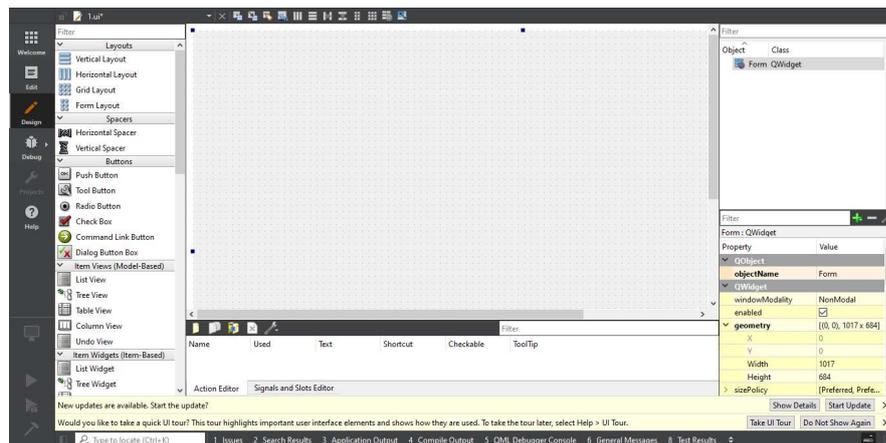


Figura 4.6. Aplicación Qt Designer.

4.4.7. FUENTE DE ALIMENTACIÓN

El primer equipo con el que se han hecho pruebas es la fuente de alimentación. (Todos los datos y características de esta fuente de tensión, han sido obtenidos de su datasheet [23]).

La fuente de alimentación que se ha utilizado es la DP832 del fabricante Rigol. Esta fuente dispone de 3 salidas, dos de ellas con un voltaje máximo 30 V y corriente máxima de 3 A, con una masa para ambas, y la tercera salida que contiene una masa interna independiente, tiene un máximo de 5 V y 3 A. La potencia total es de hasta 195 W.

Tiene bajo rizado y ruido (<350 uVrms / 3 mVpp), excelente tasa de regulación lineal y tasa de regulación de carga y un rápido tiempo de respuesta transitoria.

Con esta fuente se puede realizar mediciones de voltaje, corriente y potencia, o bien suministrárselas al circuito.

Los puertos de los que dispone son: USB Host, USB Device, USB-GPIB, LAN/RS232/Digital IO.

Sus dimensiones son 239 mm x 157 mm x 418 mm con un peso de 10.35 kg.

Entre sus aplicaciones destacan: elemento de alimentación para pruebas de circuitos electrónicos de automóviles, pruebas de automatización, verificación de características de dispositivos o circuitos y solución de problemas.

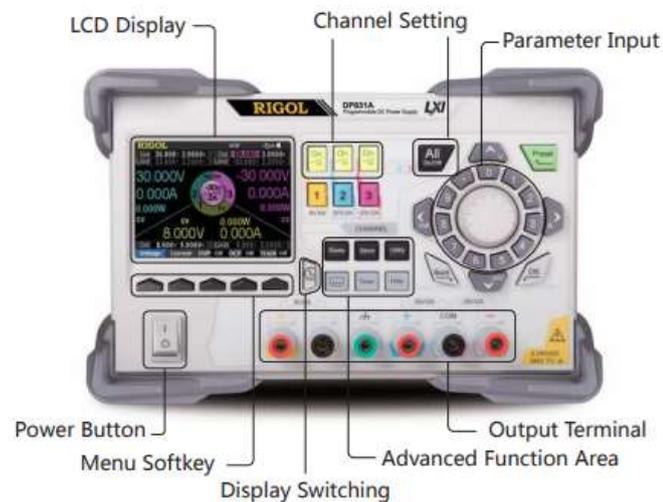


Figura 4.7. Fuente de alimentación similar a la utilizada. (Fuente: [23])

El siguiente equipo con el que se han hecho pruebas es el generador de funciones.

4.4.8. GENERADOR DE FUNCIONES

Se ha utilizado el generador de señales digital DG1062Z de Rigol.

(Todos los datos y características de esta fuente de tensión, han sido obtenidos de su datasheet [24]).

Este generador tiene 2 canales funcionales completos e independientes. Cuenta con una estabilidad de frecuencia de ± 1 ppm y baja fluctuación de 200 ps.

Tiene incorporado un generador de armónicos de 8 órdenes incorporado y un contador de 7 dígitos incorporado hasta 200 MHz.

Soporta todas las modulaciones: AM, FM, PM, ASK, FSK, PSK y PWM.

Puede generar distintos tipos de ondas: senoidales, pulsos, cuadradas, triangulares... Además de funcionar como generador de ruido o de armónicos y de modulador analógico/digital.

La máxima frecuencia que este puede aportar es de 60 MHz y la mínima de 1 μ Hz. Además de introducir la frecuencia, también se puede modificar y elegir el periodo, la amplitud, el offset y el inicio de fase. Tiene un montón de aplicaciones y de modos de uso. Es un generador muy completo con un montón de módulos de aplicación.

Las dimensiones de este son 261.5 mm \times 112 mm \times 318.4 mm y su peso es de 3.2 kg.



Figura 4.8. Generador de señales utilizado. (Fuente: [24])

El último equipo utilizado para este proyecto es un osciloscopio digital.

4.4.9. OSCILOSCOPIO

Se ha utilizado el generador de señales digital MSO1074Z-S de Rigol.

(Todos los datos y características de esta fuente de tensión, han sido obtenidos de su datasheet [25]).

Este es un osciloscopio digital de alto rendimiento y económico. Está equipado con 16 canales y permite a los usuarios medir señales analógicas y digitales al mismo tiempo.

Como se ha dicho anteriormente, este osciloscopio cuenta con 16 canales digitales y 4 canales analógicos.

El ancho de banda de los canales analógicos es de 100 MHz, 70 MHz y 50 MHz.

Tiene una velocidad de muestreo en tiempo real de hasta 1 GSa/s y una profundidad de memoria de hasta 24 Mpts. La velocidad de captura de formas de onda es de hasta 30000 wfms/s.

El rango de escala vertical es de 1 mV/div a 10 V/div. Contiene varias interfaces: USB Host y USB Device, LAN (LXI) y AUX.

El osciloscopio es de tamaño compacto, peso ligero y fácil de usar, con una pantalla LCD de 7 pulgadas con color de intensidad graduada. Concretamente sus dimensiones son 313.1 mm × 160.8 mm × 122.4 mm y pesa 3.2 kg.

Las funciones de grabación y reproducción de formas de onda en tiempo real por hardware hacen que se puedan grabar 60000 fotogramas. El disparo y decodificación de los canales analógicos y digitales se realiza al mismo tiempo.

La visualización y el análisis están correlacionados en el tiempo para las formas de onda de los canales analógicos y digitales. Permite la grabación y reproducción de formas de onda en tiempo real.

Además de representar las ondas de los cuatro canales disponibles, este osciloscopio nos permite hacer mediciones tanto de frecuencia, voltaje, valor eficaz medio... como realizar operaciones matemáticas con este.

Se puede cambiar el tipo de sonda utilizada, invertir la señal e incluso elegir la fuente que se quiere utilizar, ya que dispone de dos fuentes integradas en este.

Con el osciloscopio se puede parar la onda para hacer mejor las mediciones como volverlo a poner en marcha. Tiene una función de AUTO la cual ayuda a

detectar la onda y que las escalas y demás funciones se ajusten a la forma de onda generada.

Una función muy importante es la que nos permite adquirir datos de forma de onda que se almacenan en la memoria del osciloscopio.

Las ruletas que contiene sirven para ajustar las posiciones horizontal y vertical de la pantalla, para ver el fragmento de onda que se quiera visualizar o estudiar en cada momento. También nos permiten ajustar la posición del trigger (esto sirve mejorar la gestión de la base de datos).

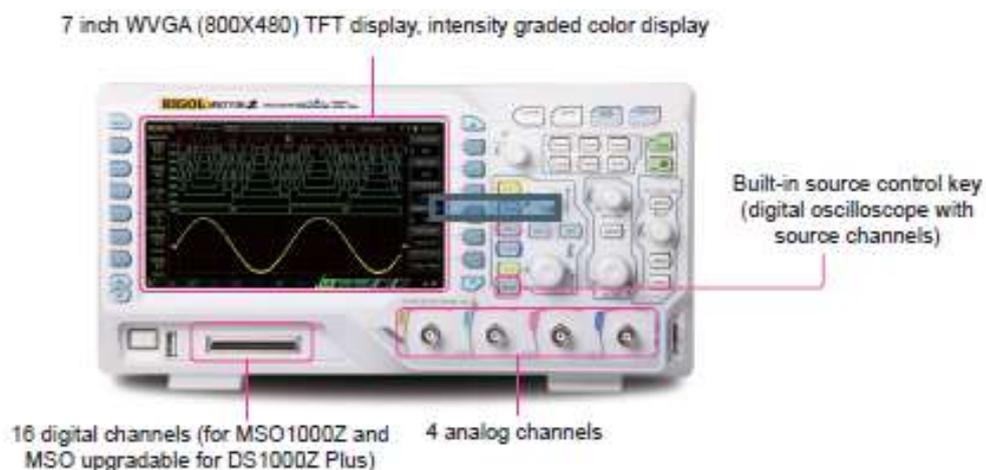


Figura 4.9. Osciloscopio utilizado. (Fuente: [25])

4.4.10. ICL7660

En relación al montaje realizado para la verificación de las interfaces, se ha utilizado un microchip, el ICL7660.

A continuación, se van a destacar las características más importantes del circuito, las cuales han sido obtenidas de su datasheet [26].

El ICL7660 es un circuito monolítico que ofrece un montón de ventajas de rendimiento sobre los dispositivos anteriores a este.

Este regulador de tensión de conmutación se emplea en electrónica de potencia, (esta es la parte de la electrónica encargada del estudio de dispositivos,

circuitos, sistemas y procedimientos para transformar y controlar voltajes y corrientes de niveles significativos, mediante el procesamiento, control y conversión de la energía eléctrica). Un regulador de tensión transforma el voltaje, el ICL7660, en concreto, invierte la señal de tensión. Los reguladores de tensión pueden ser integrados, conmutados, electromecánicos...

Este chip se ha adquirido del fabricante Renesas que es líder a nivel mundial de microcontroladores, productos analógicos, de potencia, y otras plataformas integradas.

Este circuito realiza conversiones de tensión de alimentación de positivo a negativo, para un rango de entrada de 1.5 V a 10 V, lo que da como resultado de salida complementarias de -1.5 V a -10 V, dependiendo del voltaje de entrada que se suministre.

Para el funcionamiento de este solo harían falta dos condensadores no críticos, y de esta forma, ya podría funcionar de bomba de carga y de depósito de carga.

Este chip contiene un regulador de corriente continua, un oscilador RC, un traductor de nivel de tensión y cuatro interruptores MOS de salida.

Un elemento lógico único detecta la tensión más negativa en el dispositivo y asegura que la salida de las uniones de la fuente y del interruptor de un canal en concreto, no estén hacia adelante.

El oscilador, sin carga, oscila a una frecuencia nominal de 10 kHz para una tensión de entrada de 5,0 V. Esta frecuencia se puede reducir añadiendo un condensador externo al terminal "OSC", sobrealimentado el oscilador por un reloj externo.

El terminal "LV" puede conectarse a tierra para puentear el regulador interno en serie y mejorar el funcionamiento a baja tensión. A tensiones medias y altas (de +3,5 V a +10,0 V) para el pin LV se deja al aire, para evitar el bloqueo del dispositivo.

El ILC7660 cuenta con 8 pines: NC, CAP+, GND, CAP-, V_{OUT}, LV, OSC Y V+. El rango de temperaturas que soporta es de 0 a 70 °C.

La principal aplicación de este es la alimentación negativa para RAMs dinámicas. Puede funcionar como fuente negativa económica, y es muy utilizada en los sistemas de adquisición de datos.

Este circuito puede ser utilizado en diferentes montajes, dependiendo del uso que se quiera dar. Más adelante se explicarán los montajes que se han realizado, concretando las características de estos y sus funciones, así como los elementos necesarios para el montaje de los mismos.

Una vez se han explicado todos los equipos y tecnologías empleadas, se procederá a la explicación del software creado. Se empezará contando cómo se ha realizado el diseño de las dos interfaces gráficas y posteriormente se explicará la realización del código programado para cada una de ellas.

4.5. INTERFAZ DEL PROGRAMA

Ya se ha visto anteriormente que las GUI se han creado con la aplicación Qt Designer.

Es muy importante saber el nombre que se le da a las variables, ya que a la hora de programar el código de funcionamiento se tendrán que usar estos mismos nombres de variables.

La primera de las interfaces nos permite el control de la fuente de alimentación y el generador de señales.

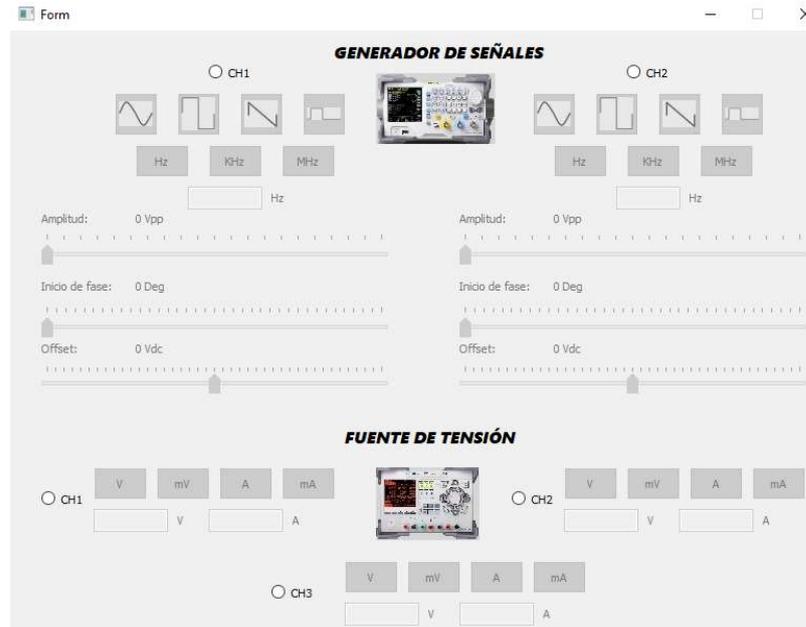


Figura 4.10. GUI Generador de señales y fuente de tensión.

Lo primero que se hizo fue añadir un Widget sobre el cual se han ido depositando los diferentes objetos de la interfaz. A este Widget se le ha limitado a un tamaño de 784 x 584 px.

Para el fondo de este widget se ha elegido un color gris claro, ya que es un color muy neutro y sobre el cual los colores destacan observándose así con facilidad. Para elegir este color, se ha tenido que programar en la propia interfaz un pequeño código en el que se define el background-color como gray.

Como se ve en la Fig.4.10 la interfaz está dividida en dos partes. La parte superior de esta la ocupa lo que se refiere al generador de señales y la parte de abajo está referida a la fuente de tensión.

Primero se ve un label que dice '*GENERADOR DE SEÑALES*', este sirve para indicar que esta parte se refiere a ese equipo. El tipo de letra y color de esta también ha sido configurado con pequeños códigos de programación.

La foto que ocupa la parte superior central es una imagen del generador de señales que se va a programar. Esta foto está incorporada en un label, modificando el tamaño de este para que la proporción de la imagen sea la correcta.

Tanto lo que se refiere al canal uno como lo que se refiere al canal dos, se ha incorporado de la misma manera por lo que se va redactar cómo se ha hecho lo referido al canal uno, teniendo en cuenta que el canal dos se hizo de forma análoga.

Lo primero que nos aparece es lo que se conoce como un '*Radio Button*' es decir, un botón de opción. Nos da la opción de pulsarlo o no, estando en color azul y con un círculo en su interior cuando este está pulsado.

A este botón se le ha añadido la opción '*checkable*' para que este se quede pulsado en el momento en el que se hace clic, y no se quitará hasta que no se vuelva a hacer clic.

Todos los botones que pertenecen a un mismo canal están incorporados dentro del mismo '*Group Box*' para que, si todos los objetos que están dentro de este tienen desactivada la función '*enabled*', luego, a la hora de programar, sólo pueden ser usados si está activado el botón de encendido del canal en el que se encuentran (dependiendo de a qué grupo se haga referencia).

Lo siguiente que se encuentran son 4 botones que servirán para elegir la forma de onda que se desea. A cada uno de estos botones se les ha incorporado la foto de la forma de onda que representará cuando sea pulsado. Estos botones también tienen activada la función '*checkable*'.

Es importante que los botones con los cuales se vayan a elegir la forma de onda estén dentro de un '*Button Group*' ya que esto va a permitir que cuando se seleccione uno de ellos, sin desactivarlo, no va a dejar seleccionar otro. Esto es muy importante, ya que un generador de señales no puede proporcionar a la vez, en el mismo canal, dos formas de onda diferentes.

Justo debajo aparecen los botones de unidades de frecuencia. Estos botones al no tener activada la función '*checkable*' no hará falta crear un '*Button Group*' porque en el momento que se selecciona uno, se deselecciona otro, no se mantienen pulsados.

El recuadro que aparece es lo que se llama un '*Line Edit*' que sirve para permitir al usuario escribir, y está acompañado por un label que por defecto va a mostrar la unidad Hz (pero que después de programarlo, se va a poder modificar).

Ahora se tienen tres labels que indican la magnitud que se quiere modificar (para que sirva al usuario de guía y sepa en todo momento que está variando), acompañados cada uno de su correspondiente '*Slider*'. Al lado de cada uno de los labels con el nombre de la magnitud se encuentra otro label que mostrará el valor que se esté marcando en el '*Slider*'. Por defecto, estarán inicializados a 0.

Todos los '*Slider*' tienen activada la opción de '*tracking*' es por ello que se muestran unas rayas encima de estos, para que el usuario visualice por donde va al mover el icono triangular.

El primero de ellos corresponde a la amplitud, por lo que, sabiendo las características del generador físico utilizado, se ha determinado un máximo de 20 Vpp, y se ha inicializado en 0 (el menor valor que se ha puesto, ya que no puede tener un valor de amplitud negativo). El segundo hace referencia al inicio de fase por lo que se sabe que este tiene que tener un máximo de 360° y un mínimo de 0° (valor en el cual está inicializado).

El tercero corresponde al offset. Como se sabe, este puede ser tanto positivo como negativo, por lo que se establece un valor mínimo de -100 y un máximo de +100 Vdc. También se le ha dado un valor por defecto de 0 Vdc.

La razón por la que el máximo y el mínimo sea de ± 100 se verá cuando se explique el código del programa.

Una vez se ha explicado toda la parte superior, se procede a explicar la parte inferior, que como se ha dicho anteriormente, corresponde a la fuente de tensión.

Aquí también se puede ver al principio un label que dice 'FUENTE DE TENSIÓN', este sirve para indicar al usuario a que equipo se están refiriendo esos objetos. El tipo de letra y color de esta también ha sido configurado con pequeños códigos de programación, pero dentro de la GUI, dentro del Qt Designer.

La foto que ocupa la parte central es una imagen de la fuente de tensión que se va a programar. Esta foto está incorporada en un label, modificando el tamaño de este para que la proporción de la imagen sea la deseada.

Tanto lo que se refiere al canal uno como lo que se refiere al canal dos y al tres, se ha incorporado de la misma manera por lo que se va explicar cómo se ha hecho lo referido al canal uno, teniendo en cuenta que para los demás canales se ha realizado del mismo modo.

En primer lugar, se tiene un '*Radio Button*' con el nombre del canal al que hace referencia. Este nos da la opción de pulsarlo o no, pero en el caso de que esté pulsado, aparecerá en color azul y con un círculo en su interior.

A este botón se le ha añadido la opción '*checkable*' para que este se quede pulsado en el momento en el que se hace clic, y no se quitará hasta que no se vuelva a hacer clic.

Hay que destacar que todos los objetos incorporados después de este tienen desactivada la función '*enabled*' para que luego, a la hora de programar, sólo puedan ser usados si está activado el botón de encendido del canal al que se refieren. Todos los botones que pertenecen a un mismo canal están incorporados dentro del mismo '*Group Box*', para que lo comentado anteriormente sea posible.

Posteriormente aparecen los botones de unidades, tanto de voltaje como de corriente. Estos son los llamados '*Push Button*' que en el momento que se presionan se activan y se desactivan, es por esto que es imposible que estén activados más de uno a la vez.

Por último, se tiene dos recuadros o '*Line Edit*' en el cual el usuario escribirá el valor deseado. Acompañando a estos se encuentran un label para cada uno, que indica la unidad de medida del valor introducido. Por defecto aparecerá en V y A ya que son las unidades más comunes.

La segunda GUI que se ha realizado corresponde al manejo del osciloscopio.

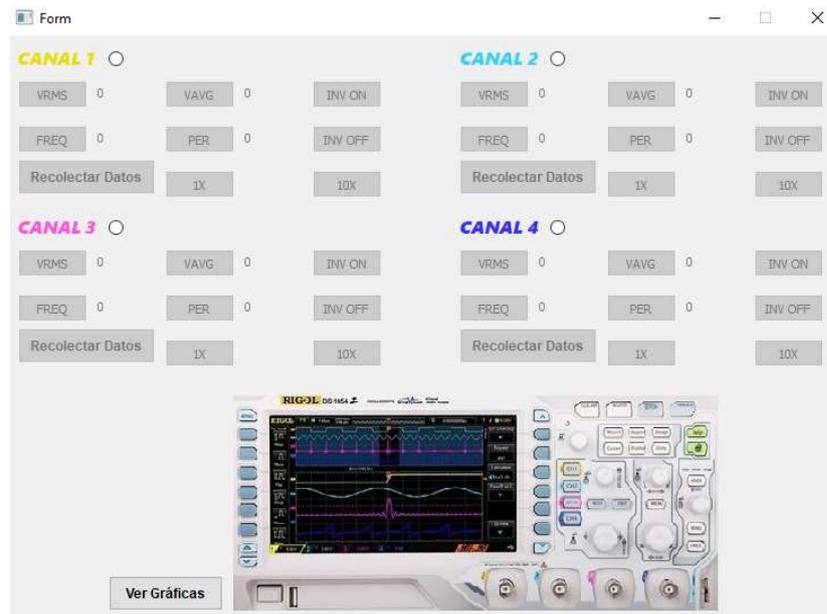


Figura 4.11. GUI Osciloscopio.

La disposición de esta interfaz consta de dos partes. En la parte inferior se dispone de un label con una foto incorporada del equipo físico que se va a programar.

Al lado de este, se encuentra un *'Push Button'* con las letras en negrita. Este botón estará siempre disponible de pulsar, sin depender de si se tiene o no activado alguno de los canales disponibles.

Por otro lado, se tiene la segunda parte de la interfaz, que ocupa toda la parte superior.

Por un lado, se tiene el label que contiene el nombre del canal (el tipo de letra, el color según el canal que sea, ya que está a juego con el del equipo físico y el tamaño de la letra, han sido configurados de forma manual, programando unas pocas líneas de código dentro de la propia GUI), y el botón de tipo *'Radio Button'* que nos da la opción de pulsarlo o no, estando en color azul y con un círculo en su interior cuando este está pulsado. Para que se quede presionado en el momento en el que es pulsado este botón, se le ha añadido la opción *'checkable'*, así se logrará también que no se deseccione hasta que no se vuelva a hacer clic. Ambos objetos siempre van a estar disponibles.

Por otro lado, se tiene 4 '*Group Box*' en los cuales están contenidos los objetos que se refieren a cada canal (hay cuatro porque el equipo cuenta con cuatro canales). Estos grupos van a permitir a la hora de programar que, si el botón del canal al que se refieren no está activo, no se va a permitir acceder a los objetos de dentro de cada grupo. Para ello, todos los objetos tienen deshabilitada la función '*enabled*'.

Cada grupo con tiene en su interior 9 botones del tipo '*Push Button*' y 4 labels que acompañan a 4 de esos botones. Estos botones tienen las mismas características, en el momento en el que se presionan se activan y se desactivan, no se mantienen pulsados ni activados.

Los cuatro labels que acompañan a cuatro de estos nueve botones, van a permitir visualizar el valor de la variable que aparecerá al pulsar el botón que le acompaña. Por defecto aparecerán en 0.

En ambas interfaces se ha programado que el ratón aparezca de una forma u otra dependiendo de dónde se encuentre. Si el ratón está en una foto de un equipo, aparecerá el símbolo de prohibido, por lo que no dejará clicar encima de esta. Si el ratón está encima de un botón simple, aparecerá una flecha y, por el contrario, si está en un botón circular o en un slider, aparecerá una mano.

Para escribir en un recuadro, el ratón adoptará la forma de escritura.

Una vez que se tienen las dos interfaces diseñadas, se guardan en la misma carpeta donde se vaya a programar el código, con el formato `.ui` acompañado del nombre que se quiera. En este caso, a la primera interfaz se le ha llamado `tfg_fyg.ui` y a la segunda `tfg_o.ui`.

El siguiente paso del proyecto es programar el código para que funcione el programa. Para esto lo primero que hay que hacer es abrir el editor de texto, en este caso PyCharm y convertir el `.ui` en `.py`, la manera de hacer esto ya ha sido explicada anteriormente.

En primer lugar, se procede a explicar el código de programación de la primera aplicación, la que gobierna a la vez la fuente de tensión y el generador.

Lo primero que se debe realizar, como en cualquier programa, es importar las librerías que se van a usar, ya que, si no, no se podrían usar determinados comandos. En Python es importante tanto el orden como las tabulaciones, ya que no dispone de ‘ ; ’ como otros. Es por esto que las librerías y la referencia al código de programación de su interfaz gráfica deben ser importadas al principio del código.

Después se necesitará definir las variables globales que se van a usar a lo largo del código, como por ejemplo las unidades de corriente, tensión, frecuencia... como las variables que servirán de multiplicador (es importantes que estas estén inicializadas a 1).

A continuación, se define una zona fija. Esta es igual para todos los programas que tengan que vincularse con una interfaz gráfica de diseñada con Qt.

Dentro de esta zona fija también se deben encontrar los comandos que identifiquen tanto a la fuente de alimentación como al generador de señales.

Estos comandos pertenecen a la librería PyVISA.

```
#INICIO ZONA FIJA
class Ventana (QtWidgets.QMainWindow):

    def __init__(self, parent=None):
        QtWidgets.QWidget.__init__(self, parent)
        self.ui = Ui_Widget()
        self.ui.setupUi(self)
```

Figura 4.12. Zona fija inicial para GUI de Qt.

Fuera ya de la zona fija, pero dentro de la función init se procederá a conectar cada botón de la interfaz gráfica (poniendo el mismo nombre que tienen en el Qt Designer) con la función que definirá la funcionalidad de ese botón.

No solo hay funciones con los botones, dentro de esta función también hay que conectar los ‘sliders’ y ‘line edit’ a su función correspondiente.

Todas las funciones que se proceden a describir tienen que estar dentro de *class Ventana*.

Las primeras funciones que hay que realizar son las que están vinculadas con los botones de encendido y apagado de cada canal y de cada equipo, tanto de la fuente como del generador.

Para esto, se vincula la característica de que el botón esté presionado con el comando SCPI referido a encender o apagar la fuente o el generador. Si es cierto que este comando varía si es el generador o la fuente, pero para cada canal, lo único que cambiaría sería el número de este.

Antes de insertar el comando SCPI necesario, tendrá que estar precedido del nombre del equipo (definido anteriormente y vinculado a la ID del instrumento físico) y `.write` ya que es una función que tiene que mandar la instrucción al equipo, no recibir información de este. Los comandos SCPI siempre irán entre paréntesis y entre comillas simples.

En estas mismas funciones es en donde se tiene que indicar si el botón está activado, y en caso de estarlo, que se activen todos los objetos que están dentro del grupo al que pertenecen.

Las siguientes funciones corresponden a las unidades de los valores que se introducen en los recuadros habilitados para ello.

Al principio de estas funciones habrá que llamar a la función global equivalente a la variable multiplicadora (por ejemplo, si la función es de kHz se llamará a la variable multiplicadora equivalente a los hercios, al igual que si la función es de MHz).

Después de esto ya podría ser utilizada sin ningún tipo de problema. Habrá que igualar esta variable al factor multiplicador correspondiente, por ejemplo, si es mA se igualará a 0.001 y si es A se igualará a 1. También habrá que indicar la unidad que es, poniendo entre comillas la unidad a la que se quiere que se refiera la función.

Una vez hecha la función, habrá que llamar a la función de corriente, de voltaje o de frecuencia del canal al que corresponda, esto dependerá del valor que se esté programando. También habrá que llamar a la variable de la interfaz gráfica

(es de tipo label) en la que tiene que aparecer la unidad del valor, y mandarle a su vez el nombre de la variable donde se tiene guardado la magnitud del valor.

En este tipo de funciones no se ha utilizado ningún comando SCPI, sólo comandos básicos de Python.

```
corrienteUnit3 = "mA"
self.ui.unidad_amperios_3.setText(corrienteUnit3)
self.function_corriente_f3()
```

Figura 4.13. Ejemplo de enviar datos a un label y de llamar a una función.

Las funciones de corriente y voltaje de la fuente de alimentación se hacen de forma análoga en los 3 canales. Primero hay que declarar las variables globales que se van a usar y crear una nueva variable que se iguale al objeto de la GUI donde se introduce el valor que se quiere enviar a la fuente.

En estas funciones es importante hacer varias condiciones:

- Si no hay ningún valor en el recuadro, se enviará a la fuente el valor 0.
- Si el valor introducido es mayor de 30 V / 30000 mV / 3 A / 300 mA (menos en el canal 3, donde el máximo valor de tensión es de 5 V o de 500 mV), se pondrá automáticamente este valor, ya que es el establecido como máximo por el fabricante.

Después de multiplicar todo esto por la variable multiplicadora que se ha definido anteriormente, se llama a una función que va a enviar los datos al equipo físico. Es importante pasarle entre paréntesis el número del canal al que se refiere la función.

La función de frecuencia es similar a anterior. En esta también hay que crear una variable que se iguale al valor que introduce el usuario en el recuadro habilitado para ello. Si esta variable nueva está vacía (no hay nada introducido), se pondrá el valor de 0 Hz. Aquí también hay que multiplicar el valor según la unidad que se haya escogido e indicarla en su correspondiente label.

Una vez se ha terminado esto, se procede a llamar al botón de la forma de onda (pulso, senoidal, cuadrada...) según cual sea el botón que esta pulsado.

Las siguientes funciones que se han creado son las correspondientes a la amplitud, el offset y el inicio de fase. Todas ellas tienen que contar con lo anterior, es decir; al terminar de definir las hay que llamar al botón de la forma de onda que esté seleccionada por el botón.

La función de la fase únicamente tiene que vincular el valor introducido en el recuadro con el label de la GUI. Este valor se tiene que almacenar en una variable global.

La función de offset aparte de realizar algo análogo a la de inicio de fase, hay que transformarla. En la interfaz se ha establecido un valor mínimo y máximo de ± 100 Vdc por lo que para corregir esto habrá que dividir el valor introducido entre 10. Se ha realizado esto para poder poner un inicio de fase decimal, y no sólo números enteros.

La función de amplitud también se hace como la de fase, pero con una condición. Esta condición es:

-Valor mínimo: $-100 + ((\text{valor introducido}) * 10 / 2)$

-Valor máximo: $+100 - ((\text{valor introducido}) * 10 / 2)$

Esto hará que el valor del offset dependa de la amplitud, siendo la suma de estos dos como mucho 20 V.

Toca ahora la realización de las funciones de las formas de onda. En estas ya se necesitan comandos SCPI. Estos comandos tienen que llevar el write entre el nombre del equipo y el SCPI ya que envían los datos que el usuario ha introducido en la GUI a los instrumentos electrónicos físicos.

En estas funciones, antes de mandar los datos, habrá que llamar a las funciones globales que tienen el valor de la frecuencia, amplitud, inicio de fase y offset para ser utilizados y enviados.

La última función que hay que definir es la función envía, que recibe de las demás funciones implicadas, el número del canal al que se refieren. Esta función sólo implica a la fuente de alimentación. Primero se llama a las funciones globales que contienen los datos ya convertidos de voltaje y corriente.

Posteriormente se utiliza el comando SCPI correspondiente, para enviar los datos introducidos, referidos a la fuente, al equipo físico.

Por último, no se puede olvidar poner otra zona fija, propia también de las GUI creadas a partir de Qt.

```
#INICIO ZONA FIJA
if __name__ == "__main__":
    app = QApplication(sys.argv)
    GUI = Ventana()
    GUI.show()
    sys.exit(app.exec_())
#FIN ZONA FIJA
```

Figura 4.14. Zona fija final para GUI de Qt.

Este primer programa tiene un total de 577 líneas de código de funcionamiento, y otras 445 referidas al diseño de la interfaz.

Una vez se ha entendido el funcionamiento de este primer programa, se procede a la explicación del código para la segunda aplicación, que permitirá el control del osciloscopio.

Este segundo programa cuenta de más librerías que el anterior.

A mayores de las importadas en el programa anterior se ha descargado e importado la librería de time (para adquirir los datos), tkinter y matplotlib (estas dos últimas son necesarias para la representación de gráficas).

Al igual que el programa anterior, lo primero que aparece en el código son las librerías empleadas y a continuación la zona fija (incluyendo los comandos PyVISA para la identificación y conexión con el equipo físico) que es exactamente igual, sólo cambia la ID del instrumento y el nombre que se le ha asociado.

Lo siguiente que aparece es la conexión de los botones con las funciones que estos realizan. En este caso sólo hay que establecer conexiones a los botones (no se dispone de ningún recuadro ni slider).

Las 4 primeras funciones permiten encender o apagar los distintos canales del osciloscopio. Primero se comprueba si el botón está presionado o no, y de ser así, se encenderá el canal y se habilitarán el resto de botones que contiene este grupo. En caso contrario el canal se apagará y se desactivarán los botones.

Para encender o apagar los canales se utilizan los comandos SCPI precedidos por el nombre que se le ha otorgado al instrumento y el `.write` para que el ordenador envíe las órdenes al equipo físico.

A continuación, se hacen las funciones que mediante el `.query` sucesivo de los comandos SCPI necesarios, se podrán obtener los valores de frecuencia, periodo, valor eficaz y valor medio (estos últimos dos valores son valores de tensión).

Todas estas funciones nos devuelven un valor en notación científica (un número multiplicado por una exponencial). Como ese número es difícil de interpretar, se ha procedido a transformarlo en un número decimal de 4 decimales (menos en la frecuencia, que sólo se le ha puesto un decimal).

La siguiente función nos permitirá invertir o no la señal. Y la siguiente es necesaria para elegir el tipo de sonda (1X o 10X). Ambas funciones sólo necesitan el comando SCPI para ejecutarlas, precedido por `.write`, ya que estas envían información al equipo físico.

Una de las funciones más importantes y con varios comandos SCPI (todos acompañados de `.write`, aunque alguno de ellos llevará ' ? ') es la de captar los datos de la forma de onda. Habrá una función de este estilo para cada canal.

En esta función se crea un fichero llamado `waveform_ch + el número del canal de la onda`. Aquí se abrirá el fichero y se creará un bucle que nos vaya escribiendo los valores de los datos de la forma de onda. El número de datos que nos proporcionará será de unos 1200 datos (se ha elegido este número porque si nos proporciona muchos valores, aparece un barrido, esto se comentará más detalladamente en el siguiente subapartado).

El fichero que nos guarda es del tipo `.csv` y se almacena en la misma carpeta en la que se encuentra el programa realizado. Al ser guardado en este formato,

nos permite abrirlo en cualquier momento y visualizarlo por ejemplo con Excel, y así representar la gráfica en cualquier momento.

Para finalizar la explicación de las funciones queda comentar la función de ver gráfica. Esta función es bastante compleja.

En primer lugar, hay que crear una ventana que nos permita acceder al ordenador y escoger cuáles serán las gráficas que se quieren visualizar. Esto nos permite visualizar varias gráficas al mismo tiempo y sobre la misma base de tiempos.

Además de abrir los ficheros y quedarse sólo con los primeros 1200 datos que posea, esta función tiene que plotearlos. Las gráficas se visualizan en una subventana ya preestablecida, que está diseñada con tkinter y que la contiene la librería anteriormente citada, matplotlib.

Además de todo esto, la función deberá leer del equipo físico (por lo que se necesitará usar `.query` más el comando necesario SCPI) la frecuencia de muestreo, ya que la escala del eje x deberá ser el resultado de dividir 1200 (puntos representados) entre la frecuencia de muestreo a la que se encuentra el osciloscopio.

También se ha establecido que los colores de las formas de onda sean los mismos que en equipo físico, dependiendo que a qué canal haga referencia, y un fondo gris y con cuadrícula, asemejándolo así a la pantalla del osciloscopio.

Por último, se necesitará, como en el caso anterior, poner la zona fija final de la misma manera que anteriormente.

Este segundo programa tiene un total de 380 líneas de código de funcionamiento, y otras 356 referidas al diseño de la interfaz.

Ambos programas, una vez se han finalizado, se ha procedido a instalar PyInstaller. Este quete hará que el usuario puede ejecutar las aplicaciones sin instalar un intérprete de Python ni ningún módulo.

A continuación, se mostrarán algunos de los comandos SCPI utilizados a lo largo del desarrollo de ambos códigos.

Estos han sido consultados y obtenidos a través de la guía de programación que proporciona el fabricante.

```

: OUTP CH <n>, {ON|OFF}
: OUTP <n>, {ON|OFF}
: CHAN <n> : DISP {ON|OFF}
: SOUR <n> : APPL: {RAMP|PULS|SQU|SIN} (valor frecuencia),
(valor amplitud), (valor offset), (valor inicio fase)
: APPL CH <n>, (voltios), (amperios)
: MEAS : ITEM? {VRMS|VAVG|PER|FREQ}, CHAN<n>
: CHAN <n> : INV {ON|OFF}
: CHAN <n> : PROB {1|10}
: WAV : SOUR : CHAN <n>
: WAV : DATA?
: ACQ : SRAT?

```

Figura 4.15. Comandos SCPI utilizados en ambos programas.

4.6. ENSAYOS REALIZADOS

En primer lugar, se comenzó el proyecto averiguando las ID de cada aparato, así como probando diferentes códigos sencillos, con los comandos SCPI, para familiarizarse con estos y ver cómo se trabajaba con ellos.

En este momento no se tenía el IDE de Pycharm, por lo que se programaron líneas sencillas de código en el editor de código simple, Sublime Text.

Todo esto que se realizó muy al principio del proyecto, fue guardado y sirvió posteriormente, para la programación del código.

Después, en lo referido a la programación del código y del diseño de la GUI, se comenzó el proyecto realizando una sola interfaz que gobernase los tres equipos a la vez.

Poco después de comenzar con la programación del código, teniendo la GUI ya lista, se empezó a barajar la opción de hacer dos interfaces por separado y que así fuesen más completas, ya que, por ejemplo, en esta primera interfaz, lo referido al osciloscopio, no disponía de la opción de medir valores (vavg, vrms, periodo o frecuencia) ni tampoco la opción de escoger una u otra sonda ni de invertir la señal.

Es por esto que se decide realizar dos interfaces más completas e independientes, pero con la característica de que puedan ser gobernadas y ejecutadas a la vez.

Cuando ya se hacen los diseños de ambas GUI, ajustándose a que dispongan del mayor número de objetos posibles para que realicen todas las funciones que son de mayor interés, se va cambiando y modelando la disposición de los objetos, los colores y los tipos de letra.

Dentro del código de funcionamiento, la primera de las interfaces (fuente de alimentación y generador de señales), no dio grandes problemas. Si que al principio no se pusieron las variables como variables globales, si no que se definían dentro de las propias funciones. Esto hizo que no funcionara, y tras estar investigando y probando diversos comandos, se llegó a la conclusión de que era porque había que declararlas como funciones globales, porque no sólo afectaban a una función, si no a varias.

El código de programación que controla al osciloscopio ha sido el más difícil y largo de programar.

Aunque dispone de menos líneas de código, tiene dos funciones que son mucho más complejas que el resto. Estas funciones son la de recogida de datos y la de ver gráficas.

En la función de recogida de datos, se barajó la opción de recoger los datos correspondientes a la amplitud uno a uno, y posteriormente los relativos al eje x, también uno a uno. Esto era una opción un poco engorrosa ya que luego, a la hora de plotearlos, no iba a poder quedar la gráfica lo más precisa posible.

Esta opción surgió debido a que, en la guía de programación del fabricante, donde se dispone de los comandos SCPI, no estaba muy bien explicado el funcionamiento del comando '*WAV DATA*' y parecía según lo que se explicaba en este, que sólo se podían captar los datos si el osciloscopio físico estaba en modo STOP.

Tras realizar varias pruebas, no se encontraba la solución para que este comando funcionara y suministrase los valores de la forma de onda. Sin embargo, al final, después de mucho buscar y probar, se encontró una solución, la cual combinada el SCPI citado anteriormente con otro comando llamado '*ACQ MDEP*'. Se realizan pruebas con este y ya no salen los errores que aparecían anteriormente.

Ahora hay otro problema. Para captar los datos, la primera opción que se realizó fue captar tantos datos como hubiera almacenados en la unidad de memoria, esto costaba un gran esfuerzo al programa y hacía que este se bloquease y dejara de funcionar. Por este motivo se cambió este dato por el que cargara 12000 puntos. Esto tampoco fue buena idea, ya que como se muestra en la Fig.4.16, al trabajar con frecuencias altas, en la gráfica quedaban los valores muy juntos.

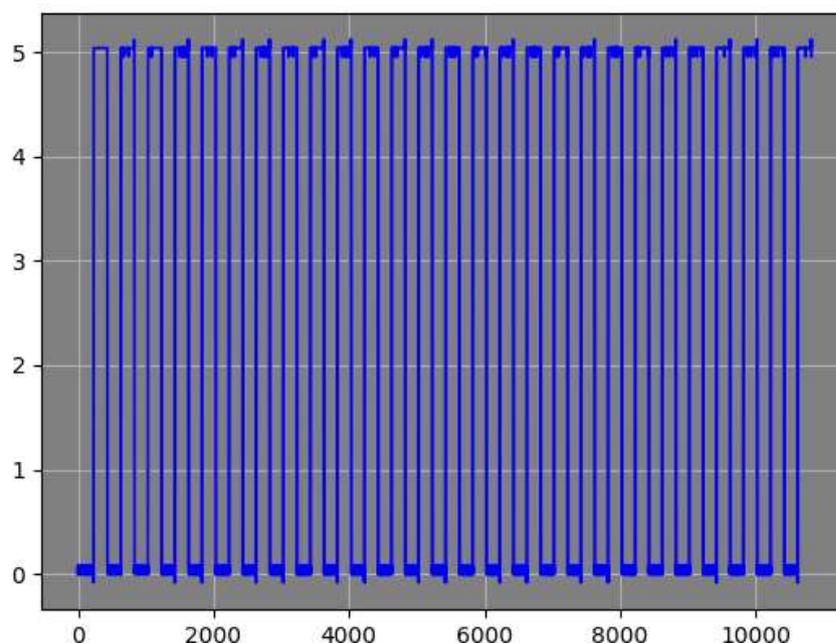


Figura 4.16. Gráfica para alto valor de frecuencia y 12000 puntos.

Es por esto que se escoge una toma de datos de aproximadamente 1200 puntos.

De esta manera se consiguió que la función funcionara correctamente.

Para que los datos se captasen de forma más clara, en el osciloscopio físico que utilizó la función adquirir, poniéndola en modo *average* en vez de en modo *normal*. En el modo *average*, el osciloscopio promedia las formas de onda de múltiples muestras para reducir el ruido aleatorio de la señal de entrada y mejorar la resolución vertical. Se consigue un mayor número de promedios que pueden reducir el ruido y aumentar la resolución vertical; mientras que, al mismo tiempo, ralentizará la respuesta de la forma de onda mostrada, a los cambios de forma de onda.

Otra de las funciones más complejas y que mayores problemas dio, es la función de ver gráfica.

Esta función dio dos problemas, ambos relacionados con la toma de datos.

El primero de ellos fue que cuando el osciloscopio nos envía los datos, hay algún valor que no es numérico, si no que contiene letras o el carácter '#', por lo que el programa no permitía visualizar los datos del fichero.

Con las primeras pruebas realizadas se observó que esos caracteres no numéricos, aparecían en la primera línea y en la última, además de en los múltiplos de 1999, por lo que en el código se quitaron los valores pertenecientes a estas líneas. Esto no funcionó.

En las siguientes pruebas que se realizaron, el programa no funcionaba, pese a los cambios que ya se habían hecho, quitando esos valores. Esto era porque cada vez aparecían caracteres no numéricos en una línea, ya no era solo en la primera, en la última y en los múltiplos de 1999, si no que ahora aparecían, por ejemplo, en la línea 35, y la siguiente vez en la 48.

Al ver esto, se cambió el código de manera que lo que hiciera fuese no coger los valores que tuvieran letras, '#' o cualquier otro carácter no numérico. De esta forma daba igual en que línea apareciera, nunca iba a coger los datos que contuvieran un carácter de este estilo.

El siguiente problema que apareció fue que al no coger 1200 puntos exactos (aunque el programa pida al osciloscopio sólo 1200 puntos, este envía algunos más por si acaso durante la transmisión, se pierden algunos), salía un problema de muestreo. Este problema no era del código ni de la GUI, si no que en el osciloscopio (en su memoria), repite el muestreo por lo que juega con el sincronismo. Es por esto que aparecía un barrido que se repetía a lo largo de la onda. Esto se ve perfectamente en la Fig.4.17.

Como se puede observar, en el momento que se pasa de 1200 puntos, nos aparece un barrido (rodeado en la Fig.4.17 de color rojo), que se repite cada aproximadamente, 1200 puntos.

Esto se debe a que cuanto menos profunda sea la memoria, más baja se mantendrá la velocidad de muestreo al captar intervalos largos de tiempo, y consecuencia de esto, aparecerá un pequeño barrido que se repite a lo largo de la función cada un cierto intervalo de puntos.

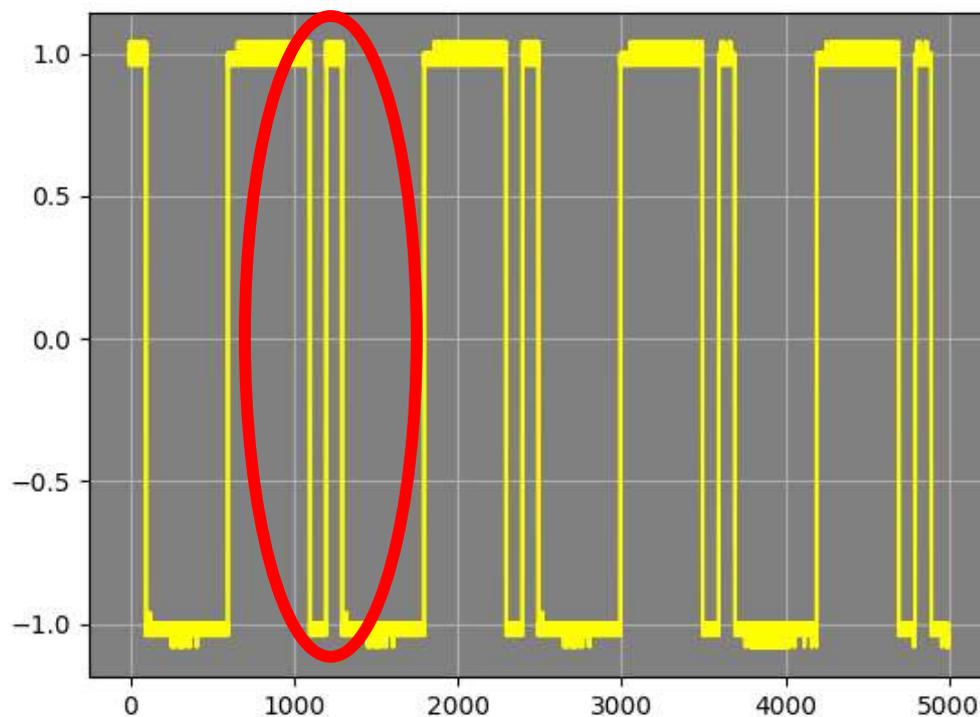


Figura 4.17. Problema de muestreo.

Esto se arregló obligando a la función a que cuando el archivo tengo más de 1200 puntos, sólo se grafiquen los 1200 puntos primeros. De esta forma, ya no saldría ese barrido.

Una vez solucionados estos problemas, y comprobado que los programas funcionan, se procedió a la realización de dos circuitos electrónicos, de aplicación en la electrónica de potencia.

Los circuitos elegidos son convertidores de continua en continua en los que el oscilador interno se sustituye por un generador de señales externo al microchip.

Se elige el montaje de este ya que los sistemas de almacenamiento de energía basados en baterías o supercondensadores pueden beneficiarse, de una forma eficiente y sencilla, de producir y almacenar energía. Los convertidores DC/DC son flexibles y con un amplio rango de trabajo en ambas direcciones, por lo que permiten ofrecer soluciones óptimas y personalizadas para cada tipo de proyecto [27].

Estos convertidores están siendo muy usados a día de hoy en las fuentes de energía renovable, ya que pueden almacenar grandes cantidades de energía de forma sencilla. Almacenar la energía permite flexibilizar la producción de energía renovable y garantizar su integración en el sistema.

A la hora de realizar estos montajes sólo se han tenido dos problemas.

Se creyó que para visualizar correctamente la forma de onda de carga y descarga del condensador, habría que poner una resistencia de 50Ω . Al realizar esto, la forma de onda de carga y descarga del condensador sí que salía más nítida (ya que el condensador se cargaba y se descargaba muy rápido) y se observaba mejor, pero, sin embargo, la señal de salida no era la esperada. Salían voltajes más elevado que los esperados ya que la resistencia que se había puesto en paralelo con el condensador, consumía corriente.

Esto se solucionó quitando esta resistencia y dejando el circuito como decía el datasheet.

El otro problema fue que, al realizar el segundo montaje, el datasheet no decía dónde poner la resistencia de $1 \text{ k}\Omega$ ni dónde colocar el generador.

Si la resistencia se ponía en el primer 7660 sólo invierte la señal (no la multiplica), por lo que esta no era la opción correcta.

Si se ponían dos resistencias en paralelo, cada una para su correspondiente integrado, y después el generador en serie con ambas, tampoco se multiplicaba la señal.

Es por esto que se optó por poner la resistencia y en serie el generador, sólo en el segundo integrado, y de esta forma, ya se obtuvo el valor de salida esperado (el valor de la entrada, pero multiplicado e invertido).

5. Resultados obtenidos

Para comprobar el funcionamiento de la interfaz, se han realizado dos montajes con el ICL7660 definido anteriormente.

El primer de ellos es un montaje simple, en el cual el oscilador del 7660 que sustituye por el generador de señales externo.

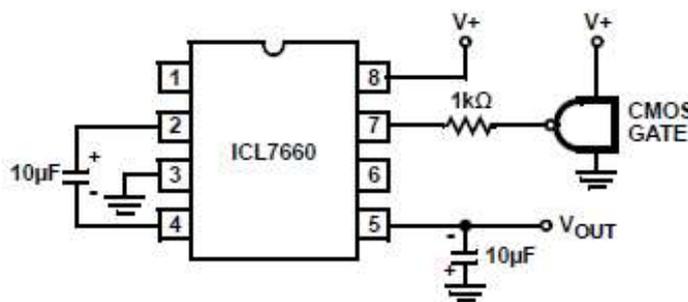


Figura 5.1. Esquema primer circuito. (Fuente: [26])

Como se puede observar en la Fig.5.1 para la realización de este sólo se ha requerido de dos condensadores de 10 μF y una resistencia de 1 $\text{k}\Omega$. Esta resistencia tiene que estar en serie con el generador para evitar posibles cortocircuitos.

Este circuito va a invertir la señal de alimentación, dando en la salida el mismo valor que a la entrada, pero negativo. Además, el valor de la frecuencia que genere la onda de salida, será la mitad del valor de frecuencia que se le indique en el generador.

El valor máximo de frecuencia de oscilación del circuito integrado es de 10 kHz.

Este circuito puede ser necesario en algunas aplicaciones (por ejemplo, cuando hay ruido) para aumentar la frecuencia del generador. Esto es logrado sobrecargando el mismo.

Hay que tener en cuenta que para valores de tensión de entrada de entre 3.5 V y 5.5 V el circuito sería el mostrado en la Fig.5.1 pero si este valor es menor de 3.5 V, habría que conectar el pin 6 del integrado, a masa.

A continuación, se muestra una foto realizada en el laboratorio, cuando se hicieron las pruebas de este circuito.

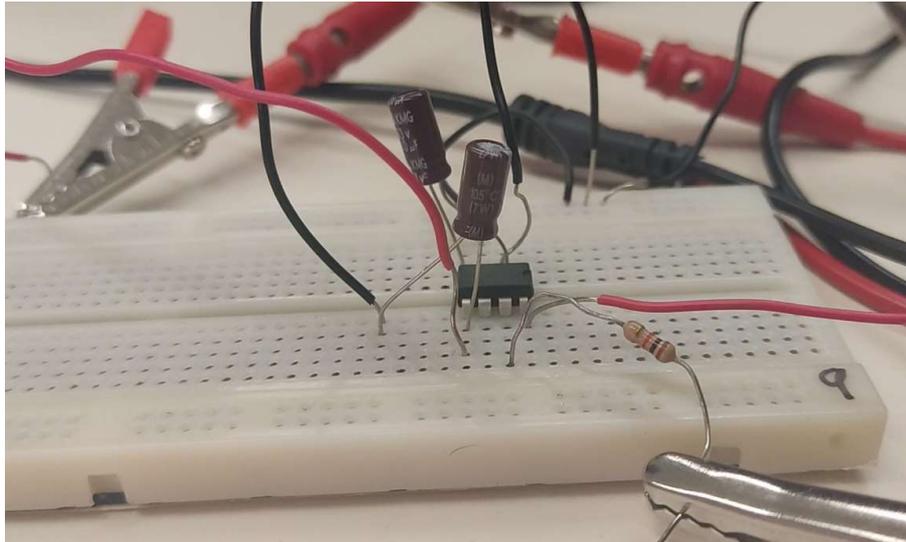


Figura 5.2. Montaje primer circuito.

Como se puede observar, una vez que se han colocado todos los elementos del circuito tal y como nos dice el datasheet, hay que conectar los tres equipos al montaje y ver qué ocurre.

Para la visualización de esto, se procede a la inserción de fotos realizadas en el laboratorio, tanto de los dos programas como del osciloscopio físico, para comprobar que la lectura de los datos y el envío de los mismos, ha funcionado correctamente en los tres equipos.

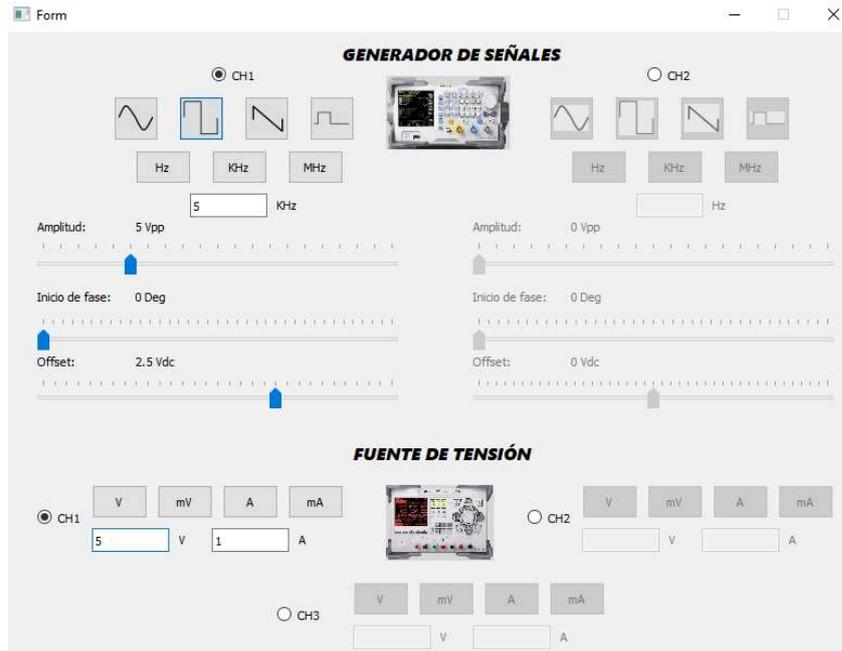


Figura 5.3. Interfaz fuente y generador del primer circuito.

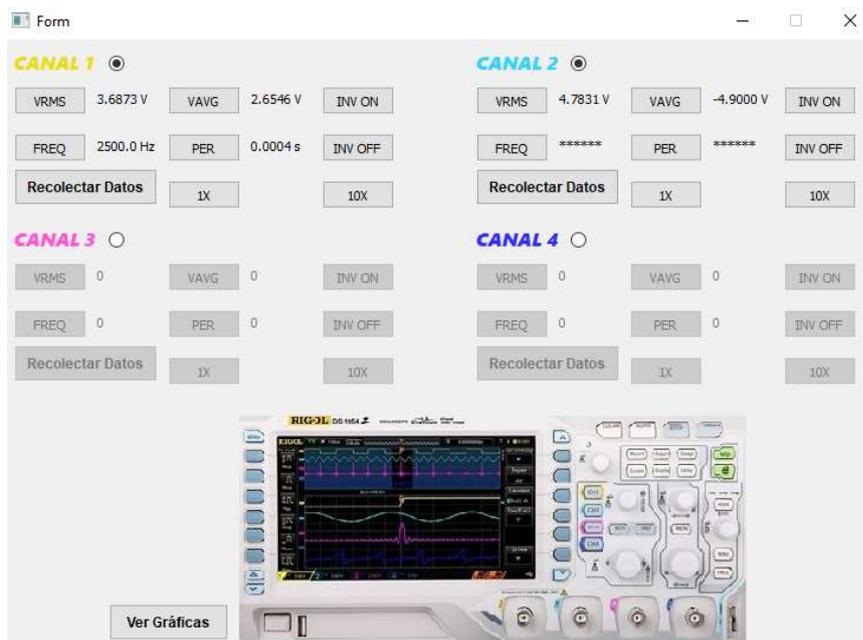


Figura 5.4. Interfaz osciloscopio del primer circuito.

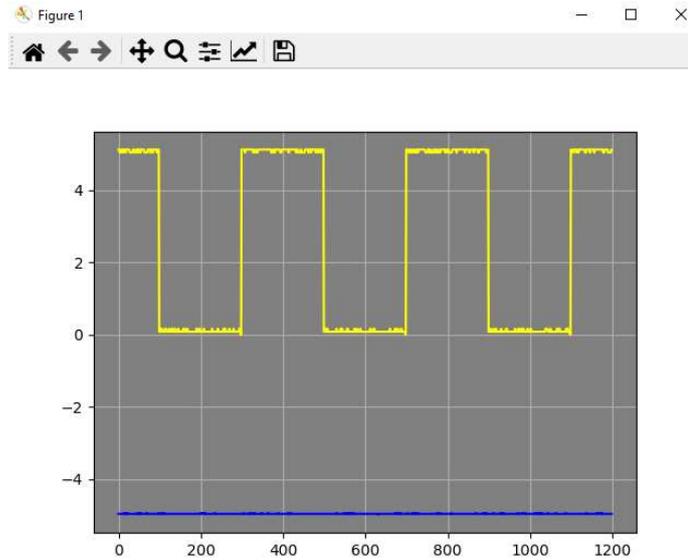


Figura 5.5. Gráfica obtenida en el primer circuito.

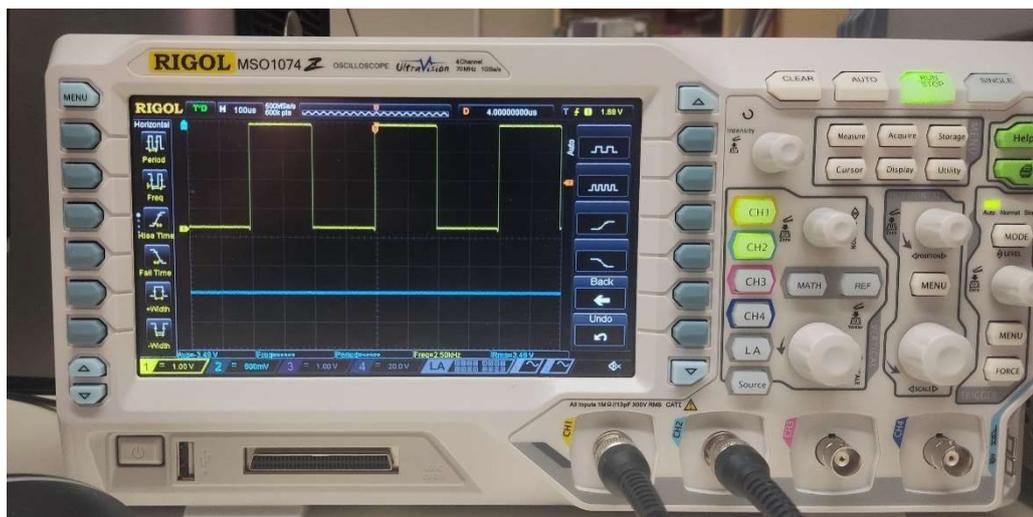


Figura 5.6. Visualización resultados del primer circuito en el osciloscopio.

Como se ve, queda demostrado el funcionamiento de la interfaz para este montaje, ya que los valores de los dos programas y de los equipos coinciden.

Cabe destacar que los ‘ * ‘ que salen en la medición de frecuencia y periodo del canal dos es porque el valor se sale de la escala del osciloscopio, ya sea por encima o por debajo.

El segundo montaje es algo más complejo que el anterior. En este se necesitan dos integrados 7660 conectados en cascada. Además, se le va a incorporar el

generador de señales, sustituyendo así el reloj interno u oscilador que estos poseen.

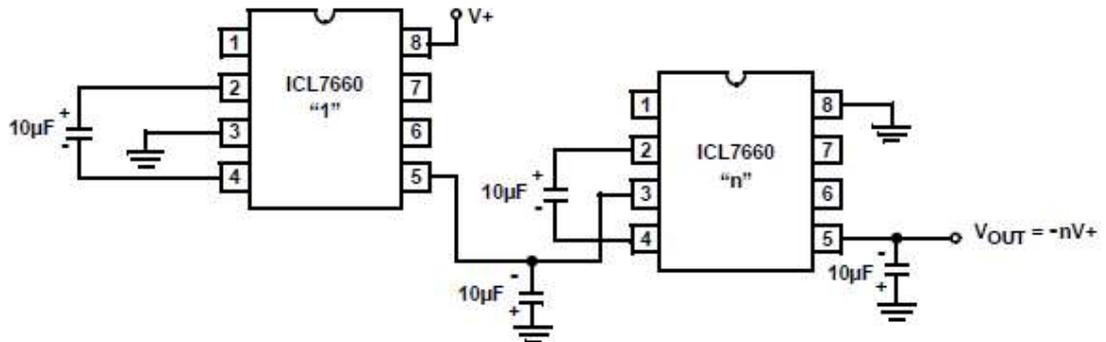


Figura 5.7. Esquema segundo circuito. (Fuente: [26])

Como se puede observar en la Fig.5.7 para la realización de este se ha requerido de cuatro condensadores de $10\ \mu\text{F}$, dos ICL7660, y a mayores, una resistencia de $1\ \text{k}\Omega$ que se conectará a la patilla 7 del segundo integrado, para conectar el generador de funciones en serie a esta.

Este circuito no sólo va a invertir la señal de alimentación, sino que va a incrementar el valor del voltaje en la salida, concretamente, va a duplicarlo. Por ello, en la salida se tendrá el doble de voltaje que, a la entrada del circuito, y con distinta polaridad. Además, el valor de la frecuencia que genere la onda de salida, será la mitad del valor de frecuencia que se le indique en el generador.

El valor máximo de frecuencia de oscilación de los circuitos integrados es de $10\ \text{kHz}$.

Este circuito funciona como duplicador negativo de la tensión de alimentación, pero, debido a la eficiencia finita de cada dispositivo, el límite práctico de dispositivos que se podrían colocar en cascada es diez ICL7660, si se trata de cargas ligeras.

Hay que tener en cuenta que para valores de tensión de entrada de entre $3.5\ \text{V}$ y $5.5\ \text{V}$ el circuito sería el mostrado en la Fig.5.7 pero si este valor es menor de $3.5\ \text{V}$, habría que conectar el pin 6 de ambos integrados, a masa.

A continuación, se muestra una foto realizada en el laboratorio, cuando se hicieron las pruebas de este circuito.

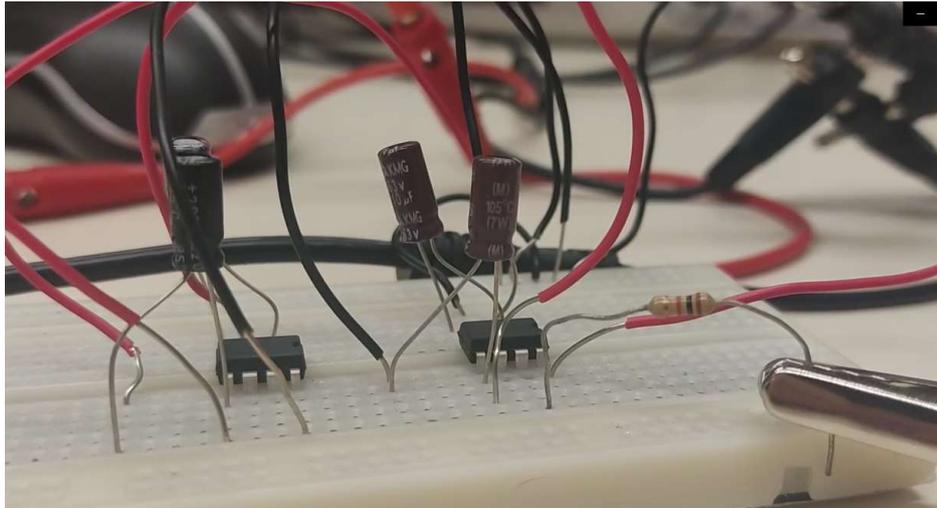


Figura 5.8. Montaje segundo circuito

Como se puede observar, una vez que se han colocado todos los elementos del circuito tal y como nos dice el datasheet, y se le ha añadido la resistencia de 1 k Ω como se ha definido anteriormente, hay que conectar los tres equipos (fuente de alimentación, generador de señales y osciloscopio) al montaje para ver qué sucede.

Para la visualización de esto, se procede a la inserción de fotos realizadas en el laboratorio, tanto de los dos programas como del osciloscopio físico, para comprobar que la lectura de los datos y el envío de los mismos, ha funcionado correctamente en los tres equipos.

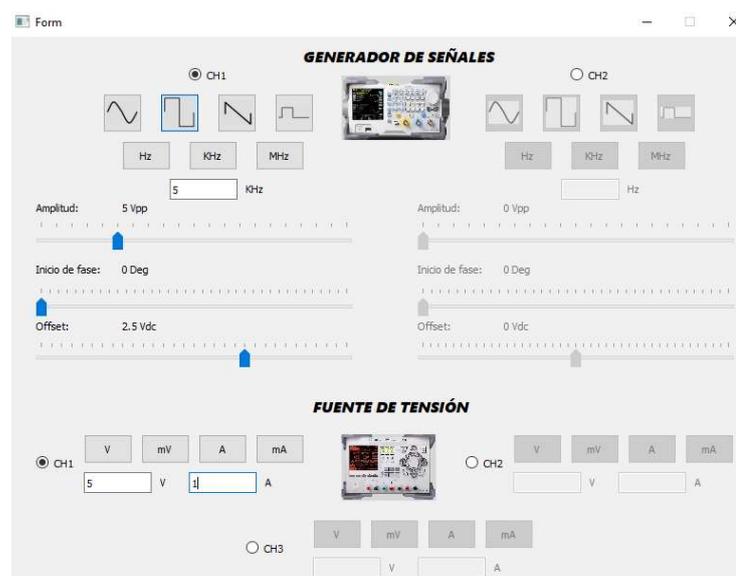


Figura 5.9. Interfaz fuente y generador del segundo circuito

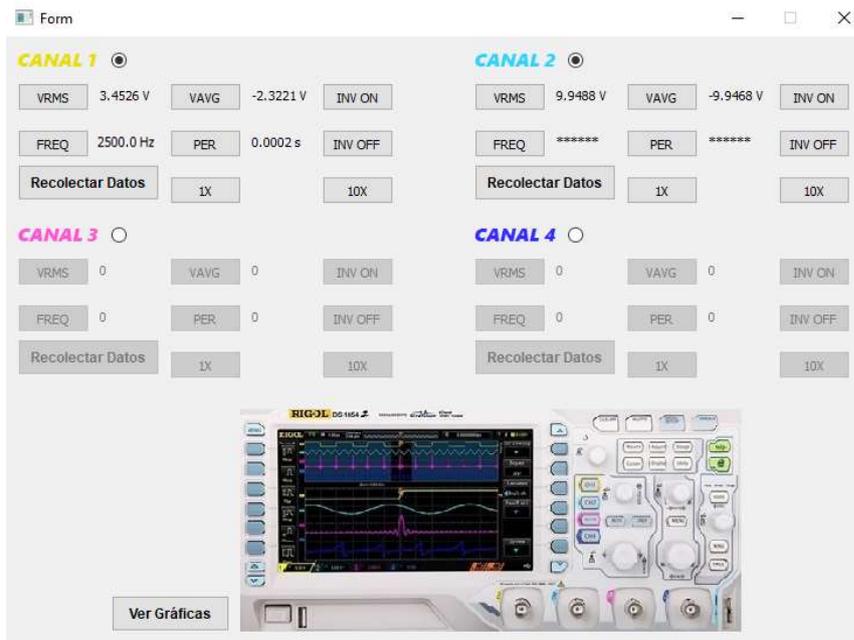


Figura 5.10. Interfaz osciloscopio del segundo circuito

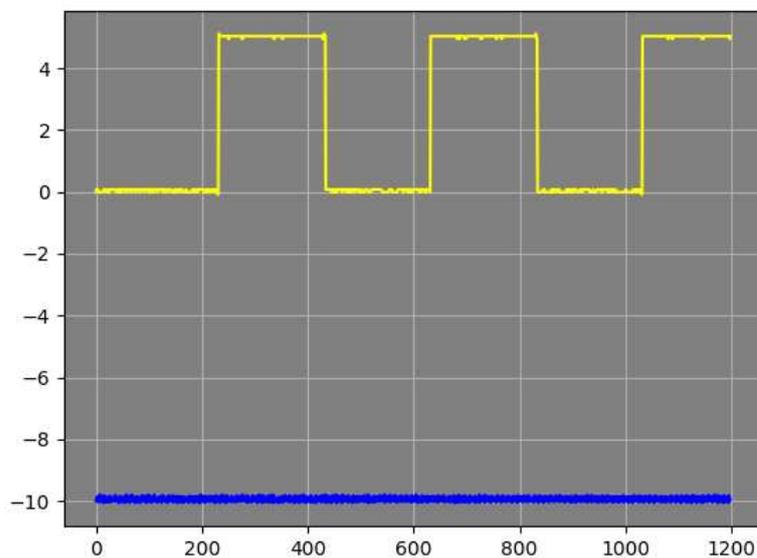


Figura 5.11. Gráfica obtenida en el segundo circuito

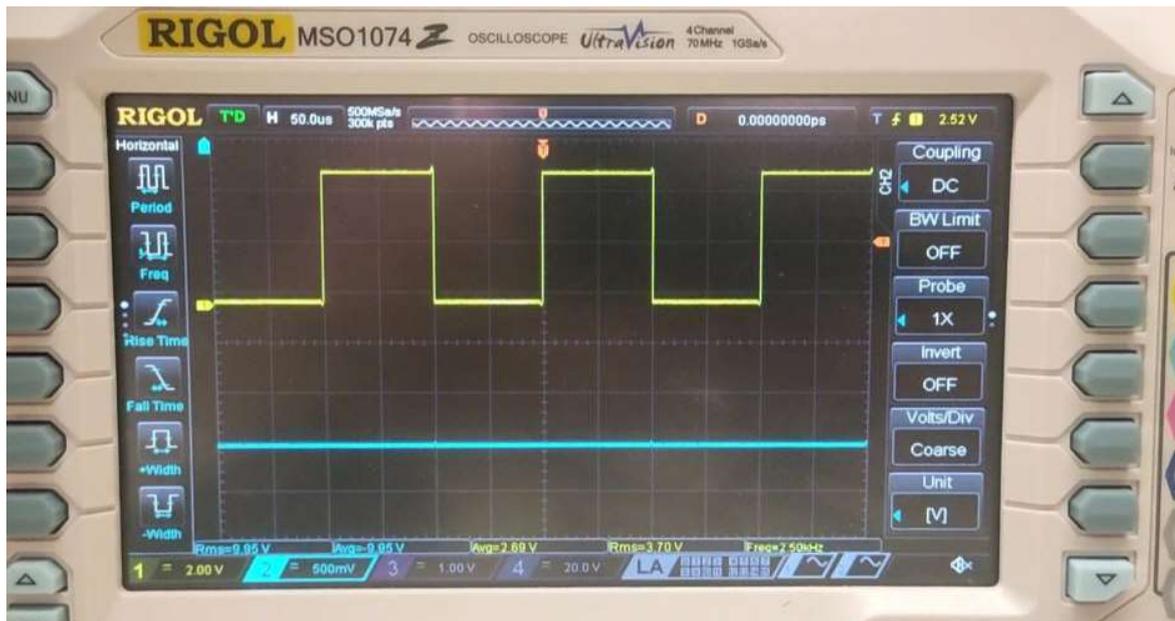


Figura 5.12. Visualización resultados del segundo circuito en el osciloscopio
 Como se ve, queda demostrado el funcionamiento de la interfaz para este montaje, ya que los valores de los dos programas y de los equipos coinciden.

6. Conclusiones y

recomendaciones

En primer lugar, resaltar que, a la hora de realizar este proyecto, se han utilizado y desarrollado diversos estudios del ámbito de la ingeniería electrónica y del campo de la automatización.

Con el desarrollo de este proyecto se ha conseguido el objetivo de demostrar que es posible controlar varios equipos electrónicos de medición de forma simultánea, mediante el uso de una interfaz gráfica y un código de programación en el lenguaje Python.

Asimismo, se verifica que el lenguaje “Python” de código abierto, es capaz de proporcionar infinidad de librerías que solucionan muchos de los problemas existentes a la hora de programar. Siendo una herramienta que permite llevar a cabo el control remoto de diferentes equipos electrónicos.

El aspecto más destacado de desarrollar la automatización de estos equipos (fuente de alimentación, generador de señales y osciloscopio, todos ellos del fabricante Rigol), es el tiempo ahorrado en realizar las distintas mediciones; realizándose estas únicamente presionando un botón, mientras que, con equipos físicos, se necesita realizar varias configuraciones hasta obtener el valor de medición buscado.

Otro de los objetivos buscado y constatado, es la posibilidad de obtener las mediciones de forma remota, y poder visualizar las gráficas y datos obtenidos en cualquier momento independientemente de si en ese instante el equipo está conectado al ordenador o no, con la ventaja que esto supone.

Tal y como se ha explicado a lo largo del proyecto, para la consecución de los objetivos marcados, se han realizado diferentes montajes basados en estudios de electrónica de potencia, utilizando reguladores de tensión que dan lugar a un convertidor de corriente continua a corriente continua, que como se ha citado, es de gran utilidad a la hora de almacenar energía.

Este proyecto podrá ser empleado en el futuro mejorando o modificando la programación a medida que cambian las necesidades funcionales y operativas del usuario, ya que el control de equipos empleado permite una gran flexibilidad al estar definido por software y no por hardware.

Una vez finalizado el proyecto y evaluados los resultados obtenidos, los nuevos objetivos a afrontar en un futuro podrían ser el ampliar el código de programación para que fuera posible visualizar las gráficas de las formas de onda en tiempo real, y, además, poder realizar la automatización con otros equipos como son el multímetro digital o la carga electrónica.

Referencias

- [1] J. M. López, «ThinkBig,» [En línea]. Available: <https://blogthinkbig.com/del-texto-a-las-ventanas-como-llegamos-a-la-interfaz-grafica-de-usuario#:~:text=Las%20primeras%20investigaciones%20sobre%20una,objetos%20o%20la%20presi%C3%B3n%20l%C3%A1ser..> [Último acceso: 2022 Junio].
- [2] «Electrónica Online,» [En línea]. Available: <https://electronicaonline.net/componentes-electronicos/>. [Último acceso: Junio 2022].
- [3] «Electronicasi,» [En línea]. Available: <http://www.electronicasi.com/enseanzas/electronica-elemental/electronica-basica/instrumentos-de-laboratorio/>. [Último acceso: Junio 2022].
- [4] «Materiales de laboratorio,» [En línea]. Available: <https://materialeslaboratorio.com/electronica/>. [Último acceso: Junio 2022].
- [5] C. Commons, «Ciencia y cultura la máximo,» [En línea]. Available: <https://www.negocioscontralaobsolescencia.com/taller/diferenciando-entre-electronica-analogica-y-digital-para-dummies-extremos>. [Último acceso: Junio 2022].
- [6] S. Zavala, «Solo es ciencia,» [En línea]. Available: <https://soloesciencia.com/2020/04/17/electronica-analogica-vs-digital/#:~:text=La%20diferencia%20entre%20la%20electr%C3%B3nica,es%20representativo%20de%20distintas%20amplitudes..> [Último acceso: Junio 2022].
- [7] A. Rubio, «Instrumentación digital,» [En línea]. Available: <https://www.instrumentaciondigital.es/diferencias-entre-instrumentacion-digital-y-instrumentacion-analogica/>. [Último acceso: Junio 2022].
- [8] «Rohde Schwarz,» [En línea]. Available: https://www.rohde-schwarz.com/sg/driver-pages/remote-control/automation-by-remote-control-step-by-step_231238.html. [Último acceso: Junio 2022].
- [9] A. Aragonés, «Control de un multímetro digital mediante python,» [En línea]. Available: <https://upcommons.upc.edu/bitstream/handle/2117/350403/tfg-ana-aragones.pdf?sequence=1&isAllowed=y>. [Último acceso: Junio 2022].

- [10] A. P. Olivares, Instrumentación virtual. Fundamentos de programación gráfica con LabVIEW, Monterrey, México: Digital, 2013.
- [11] A. A. Huertos, «Computer hoy,» [En línea]. Available: <https://computerhoy.com/reportajes/tecnologia/historia-lenguajes-programacion-428041>. [Último acceso: Junio 2022].
- [12] J. Pastor, «Xataka,» [En línea]. Available: <https://www.xataka.com/historia-tecnologica/legendario-lenguaje-programacion-cobol-acaba-cumplir-60-anos-probable-que-cumpla-otros-60>. [Último acceso: Junio 2022].
- [13] T. School, «Tokio,» [En línea]. Available: <https://www.tokioschool.com/noticias/historia-python/>. [Último acceso: Agosto 2022].
- [14] «1000 Marcas,» [En línea]. Available: <https://1000marcas.net/python-logo/>. [Último acceso: Agosto 2022].
- [15] U. d. Holguín, «Ciencias Holguín,» [En línea]. Available: <https://www.redalyc.org/pdf/1815/181531232001.pdf>. [Último acceso: Agosto 2022].
- [16] R. K. Coding, «Keep Coding,» [En línea]. Available: <https://keepcoding.io/blog/que-es-pypi/>. [Último acceso: Agosto 2022].
- [17] A. M. Díaz, «Universidad de Sevilla,» [En línea]. Available: <https://idus.us.es/bitstream/handle/11441/93787/TFG-2662-MILLAN.pdf?sequence=1&isAllowed=y>. [Último acceso: Agosto 2022].
- [18] «Digital Guide IONOS,» [En línea]. Available: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-una-gui/>. [Último acceso: Agosto 2022].
- [19] «Wikipedia, the free encyclopedia,» [En línea]. Available: https://en.wikipedia.org/wiki/National_Instruments. [Último acceso: Agosto 2022].
- [20] «PyVISA,» [En línea]. Available: <https://pyvisa.readthedocs.io/en/latest/>. [Último acceso: Agosto 2022].
- [21] «Hmong,» [En línea]. Available: https://hmong.es/wiki/Standard_Commands_for_Programmable_Instrumentation. [Último acceso: Agosto 2022].
- [22] L. P. Ramos, «Real Python,» [En línea]. Available: <https://realpython.com/qt-designer-python/>. [Último acceso: Agosto 2022].

- [23] R. Technologies, «Core Electronics,» [En línea]. Available: <https://core-electronics.com.au/attachments/DG1000Z.pdf>. [Último acceso: Agosto 2022].
- [24] R. Technologies, «Rapid,» [En línea]. Available: https://static.rapidonline.com/pdf/561770_v1.pdf. [Último acceso: Agosto 2022].
- [25] I. Americas, «Renesas,» [En línea]. Available: <https://www.renesas.com/eu/en/document/dst/icl7660-datasheet>. [Último acceso: Agosto 2022].
- [26] «Epic Power,» [En línea]. Available: <https://epicpower.es/industrias/ees/>. [Último acceso: Agosto 2022].
- [27] «Tool boom,» [En línea]. Available: <https://toolboom.com/es/articles-and-video/rigol-the-way-to-success/#:~:text=Es%20una%20compania%20nacional%20registrada,esta%20registrada%20en%20China%2C%20EE..> [Último acceso: Agosto 2022].